



CSR Synergy Framework 3.1.0

Ifconfig

API Description

August 2011



Cambridge Silicon Radio Limited

Churchill House
Cambridge Business Park
Cowley Road
Cambridge CB4 0WZ
United Kingdom

Registered in England and Wales 3665875

Tel: +44 (0)1223 692000

Fax: +44 (0)1223 692001

www.csr.com

Contents

1	Introduction.....	4
1.1	Introduction and Scope	4
1.2	Assumptions.....	4
2	Description.....	5
2.1	Introduction.....	5
2.2	Reference Model	5
3	Interface Description.....	6
3.1	Subscription.....	6
3.2	Change Interface Configuration	6
4	CSR IP Ifconfig Primitives.....	8
4.1	CSR_IP_IFCONFIG_SUBSCRIBE	9
4.2	CSR_IP_IFCONFIG_UNSUBSCRIBE	10
4.3	CSR_IP_IFCONFIG_UP	11
4.4	CSR_IP_IFCONFIG_DOWN	12
4.5	CSR_IP_IFCONFIG_UPDATE	13
4.6	CSR_IP_IFCONFIG_NAT	14
4.7	CSR_IP_IFCONFIG_FORWARD	15
4.8	CSR_IP_IFCONFIG_ARP_ENTRY_ADD.....	16
5	Document References.....	17

List of Figures

Figure 1: The CSR Ifconfig API shown relative to the TCP/IP stack	5
Figure 2: An application subscribes to CSR IP Ifconfig	6
Figure 3: The application sets the interface in UP/DOWN state	7

List of Tables

Table 1: List of CSR IP Ifconfig Primitives	8
Table 2: CSR_IP_IFCONFIG_SUBSCRIBE Primitives	9
Table 3: CSR_IP_IFCONFIG_UNSUBSCRIBE Primitives	10
Table 4: CSR_IP_IFCONFIG_UP Primitives	11
Table 5: CSR_IP_IFCONFIG_DOWN Primitive.....	12
Table 7: CSR_IP_IFCONFIG_UPDATE_IND Primitive.....	13
Table 7: CSR_IP_IFCONFIG_NAT Primitives	14
Table 7: CSR_IP_IFCONFIG_FORWARD Primitives	15
Table 7: CSR_IP_IFCONFIG_ARP_ENTRY_ADD Primitives	16
Table 8: Abbreviations and Definitions.....	18

1 Introduction

1.1 Introduction and Scope

This document describes the API between a network manager application and CSR IP. The API is called CSR IP Ifconfig. It handles administrative tasks related to the CSR IP.

The IP stack provided by CSR is the lwIP stack developed by Adam Dunkels at the Computer and Networks Architectures (CNA) lab at the Swedish Institute of Computer Science (SICS) but through the CSR API abstraction this IP stack can be replaced by any other IP stack on a target device.

1.2 Assumptions

The following assumptions and preconditions are made in the following:

- Only one instance of CSR IP is active at any time
- CSR IP Socket API is used for data communication

2 Description

This section will briefly describe the purpose of introducing the CSR IP Ifconfig API. After this section the reader should be familiar with the location of CSR IP Ifconfig API in the overall architecture and the reason for introducing the API.

2.1 Introduction

The CSR IP Ifconfig API provides a configuration interface to CSR IP.

API provides the following functionality:

- Notifying subscribed application(s) on changes to the interface(s).
- Notifying subscribed application(s) on new interface(s).
- Setting up interfaces.
- Set interface in UP or DOWN state.
- Enabling Network Address Translation (NAT) on an interface.
- Controlling whether the IP stack should perform forwarding of IP datagrams (i.e. act as a network router).

2.2 Reference Model

CSR IP Ifconfig API and its location relative to CSR IP.

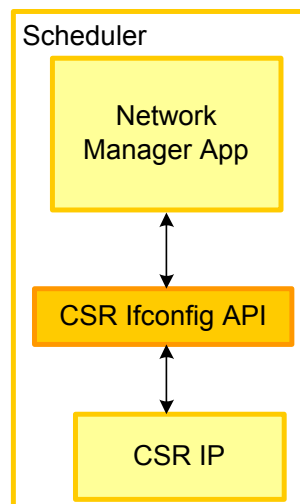


Figure 1: The CSR Ifconfig API shown relative to the TCP/IP stack

3 Interface Description

The following sessions will describe the functionality of CSR IP Ifconfig through examples using MSCs.

3.1 Subscription

Figure 2 shows that an application requests subscription on CSR IP Ifconfig by sending CSR_IP_IFCONFIG_SUBSCRIBE_REQ which is answered by a CSR_IP_IFCONFIG_SUBSCRIBE_CFM which contains "numberOfInterfaces" and there information. When an interface changes, a CSR_IP_IFCONFIG_UPDATE_IND is send to all subscribed applications, containing the updated information to the given interface.

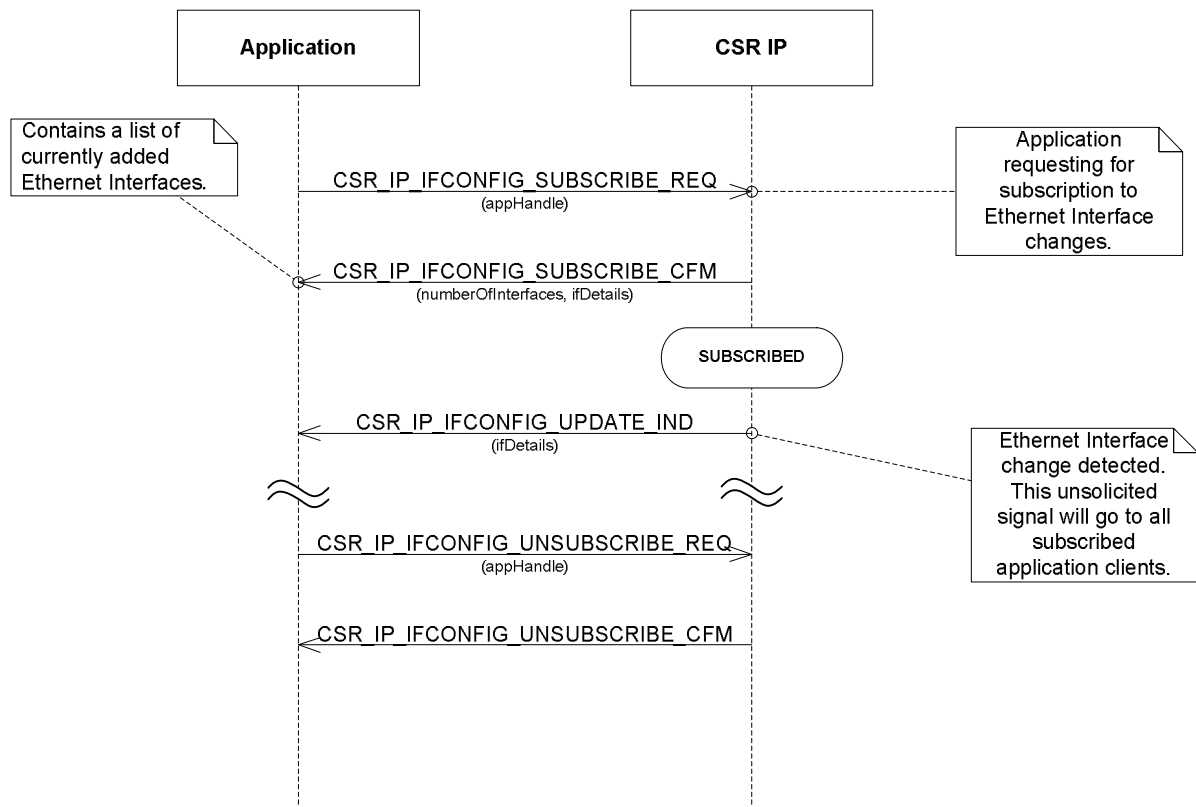


Figure 2: An application subscribes to CSR IP Ifconfig

3.2 Change Interface Configuration

Figure 3 illustrates how the Application subscribes and later configures a given interface by sending a CSR_IP_IFCONFIG_UP_REQ which contains a "configMethod" that can be either; CSR_IP_IFCONFIG_CONFIG_METHOD_STATIC for static configuration of ip address, in which case ipAddr should be set to a valid ip address, CSR_IP_IFCONFIG_CONFIG_METHOD_AUTOIP, to use AUTOIP and CSR_IP_IFCONFIG_CONFIG_METHOD_DHCP to use DHCP. CSR_IP_IFCONFIG_UP_CFM is received when the configuration is set, or if an error occurred.

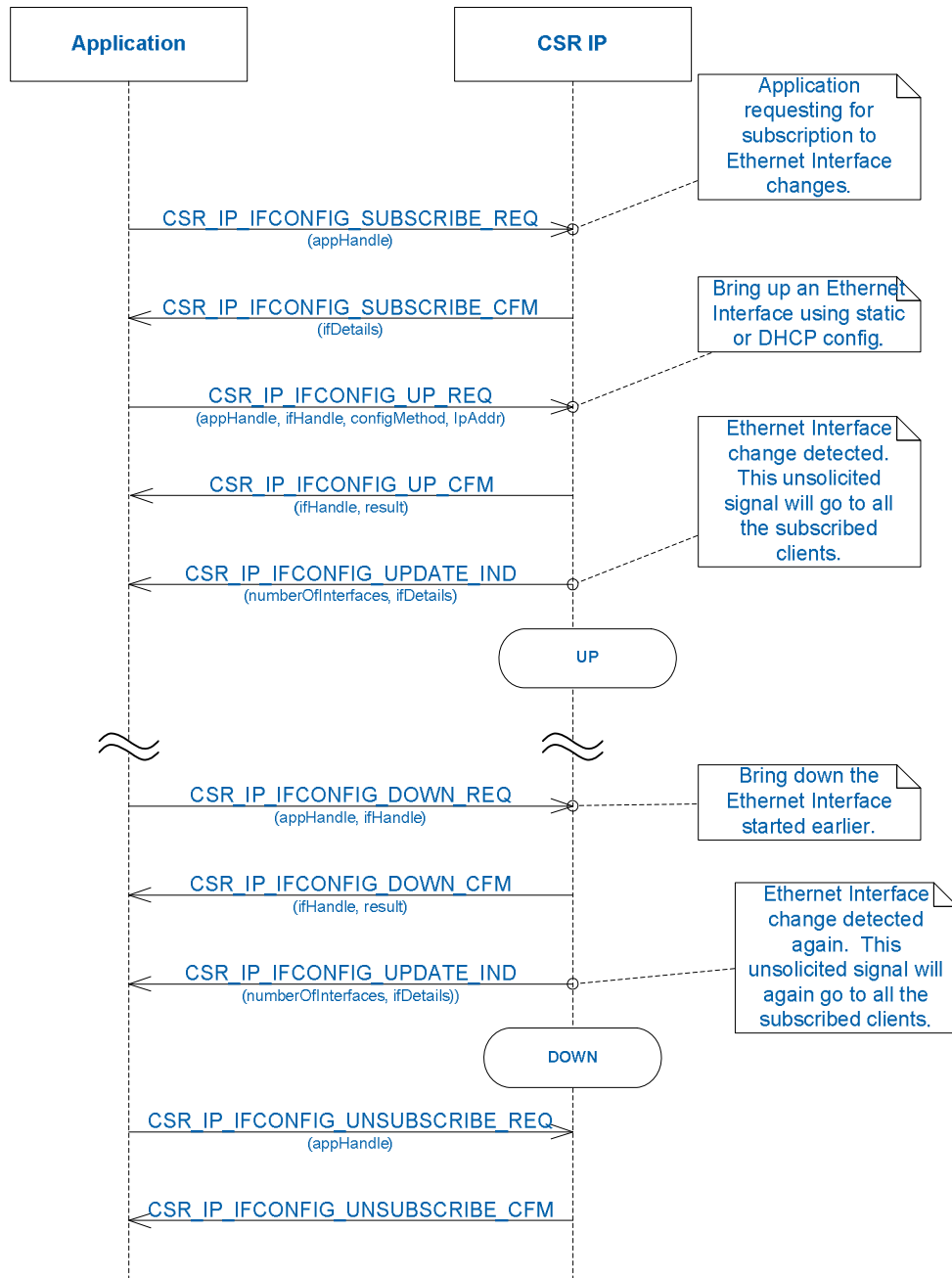


Figure 3: The application sets the interface in UP/DOWN state

4 CSR IP Ifconfig Primitives

This section gives an overview of the primitives and parameters in the interface. Detailed information can be found in the corresponding [csr_ip_ifconfig_prim.h](#) file.

Primitives	Reference
CSR_IP_IFCONFIG_SUBSCRIBE_REQ /CFM	Section 4.1
CSR_IP_IFCONFIG_UNSUBSCRIBE_REQ/CFM	Section 4.2
CSR_IP_IFCONFIG_UP_REQ/CFM	Section 4.3
CSR_IP_IFCONFIG_DOWN_REQ/CFM	Section 4.4
CSR_IP_IFCONFIG_NAT_REQ/CFM	Section 4.6
CSR_IP_IFCONFIG_FORWARD_REQ/CFM	Section 4.7

Table 1: List of CSR IP Ifconfig Primitives

4.1 CSR_IP_IFCONFIG_SUBSCRIBE

Parameters				
Primitives	type	appHandle	ifDetailsCount	*ifDetails
CSR_IP_IFCONFIG_SUBSCRIBE_REQ	✓	✓		
CSR_IP_IFCONFIG_SUBSCRIBE_CFM	✓		✓	✓

Table 2: CSR_IP_IFCONFIG_SUBSCRIBE Primitives

Description

Subscribe to CSR IP Ifconfig.

Parameters

type	CSR_IP_IFCONFIG_SUBSCRIBE_REQ/CFM
appHandle	The identity of the calling task.
ifDetailsCount	The number of interfaces
*ifDetails	Array of <code>CsrIpIfconfigIfDetails</code> for the number of interfaces

```
typedef struct CsrIpIfconfigIfDetails
{
    CsrInt32 ifHandle;
    CsrUInt8 ifType;
    CsrUInt8 ifState;
    CsrUInt8 mac[6];
    CsrUInt8 ipAddress[4];
    CsrUInt8 gatewayIpAddress[4];
    CsrUInt8 networkMask[4];
    CsrUInt8 dnsPrimaryIpAddress[4];
    CsrUInt8 dnsSecondaryIpAddress[4];
} CsrIpIfconfigIfDetails;
```

4.2 CSR_IP_IFCONFIG_UNSUBSCRIBE

Parameters	type	appHandle
Primitives		
CSR_IP_IFCONFIG_UNSUBSCRIBE_REQ	✓	✓
CSR_IP_IFCONFIG_UNSUBSCRIBE_CFM	✓	

Table 3: CSR_IP_IFCONFIG_UNSUBSCRIBE Primitives

Description

Unsubscribe from CSR IP Ifconfig.

Parameters

type	CSR_IP_IFCONFIG_UNSUBSCRIBE_REQ/CFM
appHandle	The identity of the calling task.

4.3 CSR_IP_IFCONFIG_UP

Parameters										
Primitives	type	appHandle	ifHandle	configMethod	ipAddress	gatewayIpAddress	networkMask	dnsPrimaryIpAddress	dnsSecondaryIpAddress	result
CSR_IP_IFCONFIG_UP_REQ	✓	✓	✓	✓	✓	✓	✓	✓	✓	
CSR_IP_IFCONFIG_UP_CFM	✓		✓							✓

Table 4: CSR_IP_IFCONFIG_UP Primitives

Description

Set interface in state “UP”. Notice that getting a CSR_IP_IFCONFIG_UP_CFM does not indicate that the state of the interface has changed, only that the command is understood and executed, when the interface goes “UP” a CSR_IP_UPDATE_IND is received with this information. The only valid interface state for this operation is the DOWN state. Attempting to perform this operation in the UP or MEDIA_DISCONNECTED states will return CSR_RESULT_FAILURE and CSR_IP_IFCONFIG_RESULT_INVALID_HANDLE if the interface is no longer present (state UNAVAILABLE). If the interface state changes to MEDIA_DISCONNECTED or UNAVAILABLE all configuration information pertaining to the last CSR_IP_IFCONFIG_UP_REQ is lost, and the interface state can only change to the UP state by sending another CSR_IP_IFCONFIG_UP_REQ when the interface has entered the DOWN state once again.

Parameters

type	CSR_IP_IFCONFIG_UP_REQ/CFM
appHandle	The identity of the calling task.
ifHandle	Interface handle
configMethod	CSR_IP_IFCONFIG_CONFIG_METHOD_STATIC, CSR_IP_IFCONFIG_CONFIG_METHOD_AUTOIP or CSR_IP_IFCONFIG_CONFIG_METHOD_DHCP
ipAddress	ipAddress if configMethod is CSR_IP_IFCONFIG_CONFIG_METHOD_STATIC
gatewayIpAddress	gatewayIpAddress if configMethod is CSR_IP_IFCONFIG_CONFIG_METHOD_STATIC
networkMask	networkMask if configMethod is CSR_IP_IFCONFIG_CONFIG_METHOD_STATIC
dnsPrimaryIpAddress	dnsPrimaryIpAddress if configMethod is CSR_IP_IFCONFIG_CONFIG_METHOD_STATIC
dnsSecondaryIpAddress	dnsSecondaryIpAddress if configMethod is CSR_IP_IFCONFIG_CONFIG_METHOD_STATIC
result	CSR_RESULT_SUCCESS, CSR_RESULT_FAILURE or CSR_IP_IFCONFIG_RESULT_INVALID_HANDLE

4.4 CSR_IP_IFCONFIG_DOWN

Parameters Primitives				
	type	appHandle	ifHandle	result
CSR_IP_IFCONFIG_DOWN_REQ	✓	✓	✓	
CSR_IP_IFCONFIG_DOWN_CFM	✓		✓	✓

Table 5: CSR_IP_IFCONFIG_DOWN Primitive

Description

Set interface in state “DOWN”. The only valid interface state for this operation is the UP state. Attempting to perform this operation in any other state shall be responded with result CSR_RESULT_SUCCESS as the interface is considered already down in all other states. However, if the interface is no longer available (state UNAVAILABLE) the result shall be CSR_IP_IFCONFIG_RESULT_INVALID_HANDLE.

Parameters

type	CSR_IP_IFCONFIG_UP_REQ/CFM
appHandle	The identity of the calling task.
ifHandle	Interface handle
result	CSR_RESULT_SUCCESS, CSR_RESULT_FAILURE or CSR_IP_IFCONFIG_RESULT_INVALID_HANDLE

4.5 CSR_IP_IFCONFIG_UPDATE

Parameters	type	ifDetails
Primitives		
CSR_IP_IFCONFIG_UPDATE_IND	✓	✓

Table 6: CSR_IP_IFCONFIG_UPDATE_IND Primitive

Description

Receive interface update.

Parameters

type	CSR_IP_IFCONFIG_UPDATE_IND
ifDetails	Interface information

4.6 CSR_IP_IFCONFIG_NAT

Parameters Primitives					
	type	appHandle	ifHandle	enable	result
CSR_IP_IFCONFIG_NAT_REQ	✓	✓	✓	✓	
CSR_IP_IFCONFIG_NAT_CFM	✓		✓		✓

Table 7: CSR_IP_IFCONFIG_NAT Primitives

Description

CSR_IP_IFCONFIG_NAT is used for controlling whether the IP stack performs NAT on a given network interface.

NAT is the process of rewriting sender and receiver addresses in IP datagrams to make the packets appear to originate from another host than the original sender. This is for instance used in network routers that have a number of clients which have IP addresses that can not be used for global communication on the Internet.

A router performing NAT enables such clients to communicate globally by making their packets appear to come from the router itself before passing them on in the network, thus allowing the packets to reach their destination. When packets destined for the clients are received by the router, it replaces the destination address by that of the client for which the packet is really for and passes it on to the client.

If NAT is enabled on a given interface, a packet with a source address different from the address of the network interface on which it is about to be sent out on is rewritten to have a source address of said network interface. The inverse process happens on reception: The destination address is changed from that of the network interface to the address of the client. A table is used to identify packets that belong to connections (based on source and destination IP addresses and ports) on which to perform address rewriting.

Parameters

type	CSR_IP_IFCONFIG_NAT_REQ/CFM
appHandle	Scheduler queue ID to return the confirmation message to.
ifHandle	Handle identifying the interface to enable/disable NAT on.
enable	Boolean value that indicates if NAT should be enabled or disabled.
result	Result of the request. If the IP stack does not support NAT, it will return CSR_IP_IFCONFIG_RESULT_NOT_SUPPORTED, otherwise it returns CSR_RESULT_SUCCESS if the request could be fulfilled and CSR_RESULT_FAILURE if an unspecified error occurred.

4.7 CSR_IP_IFCONFIG_FORWARD

Parameters	type	appHandle	enable	result
Primitives				
CSR_IP_IFCONFIG_FORWARD_REQ	✓	✓	✓	
CSR_IP_IFCONFIG_FORWARD_CFM	✓			✓

Table 8: CSR_IP_IFCONFIG_FORWARD Primitives

Description

Controls the behavior of the IP stack for received datagrams that are not destined for an IP address assigned to any of the local network interfaces. If enabled, the IP stack will attempt to look up a route to the destination IP address in the datagram and forward it over the appropriate network interface. If not enabled, the IP stack will silently drop the datagram.

Parameters

type	CSR_IP_IFCONFIG_FORWARD_REQ/CFM
appHandle	Scheduler queue ID to return the confirmation message to.
enable	Boolean value that indicates if forwarding should be enabled or disabled.
result	Result of the request. If the IP stack does not support forwarding of IP datagrams, it will return CSR_IP_IFCONFIG_RESULT_NOT_SUPPORTED, otherwise it returns CSR_RESULT_SUCCESS if the request could be fulfilled and CSR_RESULT_FAILURE if an unspecified error occurred.

4.8 CSR_IP_IFCONFIG_ARP_ENTRY_ADD

Parameters					
Primitives	type	appHandle	mac	ipAddress	result
CSR_IP_IFCONFIG_ARP_ENTRY_ADD_REQ	✓	✓	✓	✓	
CSR_IP_IFCONFIG_ARP_ENTRY_ADD_CFM	✓				✓

Table 9: CSR_IP_IFCONFIG_ARP_ENTRY_ADD Primitives

Description

Optional interface to add a new dynamic (non-static/-persistent) ARP cache entry.

Parameters

type	CSR_IP_IFCONFIG_FORWARD_REQ/CFM
appHandle	Scheduler queue ID to return the confirmation message to.
enable	Boolean value that indicates if forwarding should be enabled or disabled.
result	<p>Result of the request. If the IP stack does not support setting ARP cache entries, it will return CSR_IP_IFCONFIG_RESULT_NOT_SUPPORTED. Otherwise it returns CSR_RESULT_SUCCESS if the request was fulfilled, CSR_IP_IFCONFIG_RESULT_UNKNOWN_NETWORK if the requested IP address was not found to be part of a network configured of any network interfaces, and CSR_RESULT_FAILURE if an unspecified error occurred.</p> <p>CSR_IP_IFCONFIG_RESULT_UNKNOWN_NETWORK is returned e.g. if the system has a single network interface configured with an address in the 192.168.0.0/16 network and a request is made to set up an ARP entry for 192.169.0.1. 192.169.0.1 lies outside the network configured on the network interface, and as such it is meaningless to create an ARP entry for it.</p>

5 Document References

Ref	Title
-----	-------

Terms and Definitions

BlueCore®	Group term for CSR's range of Bluetooth wireless technology chips
Bluetooth®	Set of technologies providing audio and data transfer over short-range radio connections
CSR	Cambridge Silicon Radio
MSC	Message Sequence Chart
UniFi™	Group term for CSR's range of chips designed to meet IEEE 802.11 standards

Table 10: Abbreviations and Definitions

Document History

Revision	Date	History
1	22 FEB 10	Initial revision
2	30 NOV 09	Ready for release 2.0.0
3	20 APR 10	Ready for release 2.1.0
4	DEC 10	Ready for release 3.0.0
5	Aug 11	Ready for release 3.1.0

TradeMarks, Patents and Licences

Unless otherwise stated, words and logos marked with [™] or [®] are trademarks registered or owned by CSR plc or its affiliates. Bluetooth[®] and the Bluetooth logos are trademarks owned by Bluetooth SIG, Inc. and licensed to CSR. Other products, services and names used in this document may have been trademarked by their respective owners.

The publication of this information does not imply that any licence is granted under any patent or other rights owned by CSR plc.

CSR reserves the right to make technical changes to its products as part of its development programme.

While every care has been taken to ensure the accuracy of the contents of this document, CSR cannot accept responsibility for any errors.

No statements or representations in this document are to be construed as advertising, marketing, or offering for sale in the United States imported covered products subject to the Cease and Desist Order issued by the U.S. International Trade Commission in its Investigation No. 337-TA-602. Such products include SiRFstarIII[™] chips that operate with SiRF software that supports SiRFInstantFix[™], and/or SiRFLoc[®] servers, or contains SyncFreeNav functionality.

Life Support Policy and Use in Safety-critical Compliance

CSR's products are not authorised for use in life-support or safety-critical applications. Use in such applications is done at the sole discretion of the customer. CSR will not warrant the use of its devices in such applications.

Performance and Conformance

Refer to www.csrsupport.com for compliance and conformance to standards information.