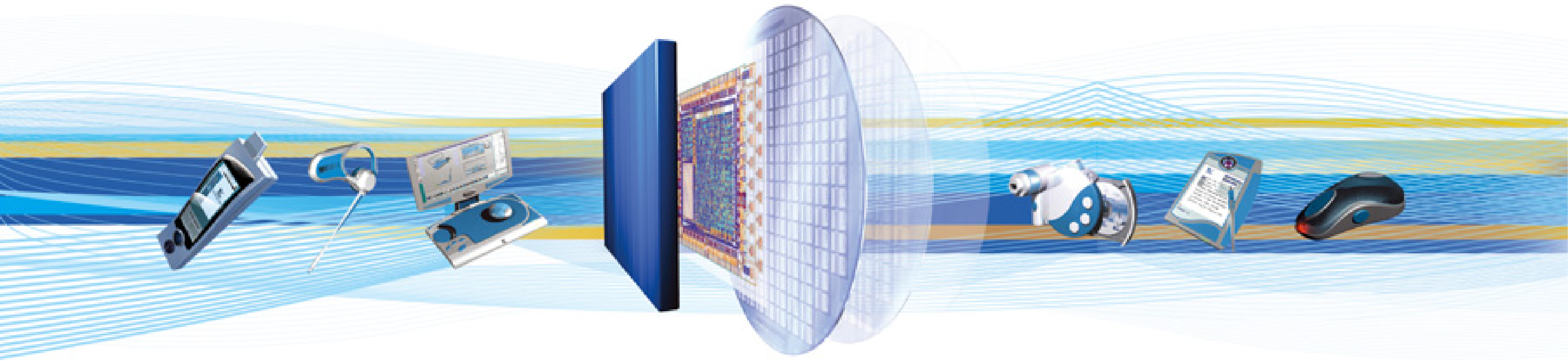
CSR Synergy®

CSR Synergy Framework 3.1.0

API Changes between 2.X and 3.X

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

This is a description of the API changes that have been introduced between CSR Synergy Framework 2.X and CSR Synergy Framework 3.X.

The intention is that this document is used in existing technologies and portings updating to 3.X framework version.

For all API changes please see the appropriate API documentation for the detailed description and order of parameters and use of the function.

The description is divided into generic api functions (GSP) and ported api functions (BSP).

2 GSP API Changes

2.1 csr_gsched

This library has been renamed to csr_sched and no longer contains an implementation of csr_pmem.h.

2.2 csr_msgconv

This interface and associated functionality has been updated to use CsrSize instead of CsrUint16 to represent sizes and offsets, in order to support serialisation and deserialisation of objects with a size larger than 65535.

The serialised format of CsrUint16 and CsrUint32 has been changed from big endian to little endian.

In addition, some redundant and/or unused functionality has been removed:

```
CrsBoolSer/Des
CsrUint24Ser/Des
CsrPrimSer/Des
CsrSchedQidSer/Des
CsrSchedTidSer/Des
```

2.3 csr_random

The csr_random.h interface and corresponding implementation has been added to the GSP replacing the existing BSP interface and implementation:

```
void *CsrRandomSeed(void);
CsrUint32 CsrRandom(void *randomState);
```

Please see the header csr_random.h for usage information. The implementation is contained in the new library csr_random. Please note that the previous BSP implementation was contained in the BSP library csr_util.

2.4 csr_unicode.h

The CsrUtf8StrTruncate function has been added to the GSP. Please see csr_unicode.h for further information.

The behaviour of CsrUtf8StrNCpyZero has changed slightly, so that it will properly truncate UTF-8 strings on a character boundary. Please see csr_unicode.h for exact behaviour.

2.5 csr_sdio

The csr_sdio_sdio and csr_sdio_cspi libraries have been renamed to csr_sdio. Use the configuration options CSR_SDIO_USE_SDIO and CSR_SDIO_USE_CSPI to control which variant is compiled into the library.

Please note that the current implementation is only capable of handling either SDIO or CSPI, not both at the same time, so only one of CSR_SDIO_USE_SDIO and CSR_SDIO_USE_CSPI can be set at the same time.

2.6 csr_log.h

CSR_LOG_LEVEL_TASK_MEMORY_* and CSR_LOG_LEVEL_ENVIRONMENT_LIST_* have been removed as they are unused and have no effect.

2.7 csr_log_text.h

CSR_LOG_TEXT_ASSERT and CSR_LOG_TEXT_UNHANDLED_PRIMITIVE macros have been added as wrappers on top of CSR_LOG_TEXT_CRITICALto facilitate easy markup of the two most common events/conditions.

2.8 csr_log_register.h

The CsrLogRegisterPlatform function and the CsrLogPlatformInformation type as well as the corresponding log formatter callback lregplatform have been removed. This information is now automatically detected by the pcap formatter and emitted to the log at the right time.

2.9 csr_log_version.h

This header has been moved from gsp/inc to gsp/src/log/inc as it is for internal use only.

2.10 csr_hci_sco.h

The following parameter is now qualified with const:

```
CsrHciLookForScoHandle - theBuf
```

2.11 csr_hci_*.h

The CSR_HCI_HOUSE_CLEANING primitive has been removed from the public interface as it was for internal use only.

2.12 csr_macro.h

The CSR_PARAM_UNUSED macro has been removed as it has the same functionality as CSR_UNUSED.

2.13 csr_data_store_*.h

The CSR_DATA_STORE_HOUSE_CLEANING primitive has been removed from the public interface as it was for internal use only.

2.14 csr_am_*.h

The CSR_AM_HOUSE_CLEANING primitive has been removed from the public interface as it was for internal use only.

2.15 csr_app_*.h

The CSR_APP_HOUSE_CLEANING primitive has been removed from the public interface as it was for internal use only.

2.16 csr_deprecated

This library has been removed along with all of the CSR Synergy Framework 1.x support code.

2.17 csr_ui_general.h

This redundant header has been removed.

2.18 csr_app_main.h

Added the function CsrAppMainUsage which must be supplied by the application to allow it to print usage information about additional command line parameters. See csr_app_main.h for more information.

CsrAppMainBluecorePsrFileGet and CsrAppMainBluecorePsrStringGet have been added to allow the application to extract the corresponding command line arguments.

3 BSP API Changes

3.1 csr_types.h

CsrPtrdiff, CsrUintptr and CsrIntptr have been added to csr_types.h and needs to be mapped to the equivalent types on the platform. The mapping to a standard conforming system is:

```
CsrPtrdiff -> ptrdiff_t
CsrUintptr -> uintptr_t or size_t
CsrIntptr  -> intptr_t or ptrdiff_t
```

Please see the ANSI/ISO C90 and/or ANSI/ISO C99 for further information about the properties of these types.

3.2 csr_sdio.h

Added priv field (void *) to CsrSdioFunctionDriver struct, for use by the SDIO Driver.

Added device field (void *) to CsrSdioFunction struct, for use by the Function Driver to determine if two functions are on the same device.

The CsrSdioCallbackInhibitEnter/Leave() functions have been removed as they are no longer needed.

3.3 csr_util.h

The CsrRandom interface and the corresponding implementation has been removed from the BSP and consequently the BSP library csr_util does not contain the definition of the CsrRandom function anymore. Please see csr_random.h in GSP for replacement.

3.4 csr_sched.h

The following parameters are now qualified with const:

```
CsrSchedMessagePutStringLog - file
CsrSchedMessageBroadcastStringLog - file
CsrSchedTimerSetStringLog - file
CsrSchedTimerCancelStringLog - file
```

3.5 csr_framework_ext.h

CsrGlobalMutexLock and CsrGlobalMutexUnlock no longer return a result. They are assumed to always succeed.

The following parameter is now qualified with const:

```
CsrThreadCreate – threadName
```

The sleepTimeInMs parameter of CsrThreadSleep is now a CsrUint16 instead of CsrUint32, to reduce the porting complexity and to match the type of the timeoutInMs parameter of the CsrEventWait function.

3.6 platform/csr_framework_ext_init.h

The platform dependent (and optional) functions CsrGlobalMutexCreate and CsrGlobalMutexDestroy have been made redundant in the supplied BSP's and have been removed.

3.7 platform/csr_log_transport_init.h

The following parameters are now qualified with const:

```
CsrLogTransportFileOpen - fileName
CsrLogTransportFileAsyncOpen - fileName
CsrLogTransportFileStaticOpen - logFileName
```

3.8 csr_serial_com.h

CsrUartDrvTx has a new function signature:

```
        CsrBool CsrUartDrvTx(void *handle, const CsrUint8 *data, CsrUint32
                             dataLength, CsrUint32 *numSent);
```

Previously `data' was of type `char *'.

CsrUartDrvReset has been removed as it was unused.

The following parameters are now qualified with const:

```
        CsrUartDrvDataRx - data
        CsrUartDrvTx - data
```

3.9 platform/csr_serial_init.h

This platform dependent (and optional) interface has been replaced with a more general interface that allows the individual serial port parameters to be set and the handle is now opaque and allocated by the driver. Two new functions replace CsrUartDrvConfigure:

```
        void *CsrUartDrvInitialise(...);
        void CsrUartDrvDeinitialise(void *handle);
```

CsrUartDrvInitialise is called to specify the serial port parameters and prepare an instance of the serial driver and returns a handle for the instance. CsrUartDrvInitialise must be called once for each required instance. If the handle returned is NULL, no further instances can be created. During shutdown, the handle must be released by a call to CsrUartDrvDeinitialise, to allow any allocated resources to be freed.

3.10 csr_usb_com.h

CsrUsbDrvReset has been removed as it was unused.

3.11 platform/csr_usb_init.h

The following parameter is now qualified with const:

```
        CsrUsbDrvConfigure - device
```

3.12 csr_ip_ifconfig_*.h

The CSR_IP_IFCONFIG_DHCP_RENEW_REQ and CSR_IP_IFCONFIG_DHCP_RENEW_CFM primitives have been removed from the API as they are unused.

The CSR_IP_IFCONFIG_FORWARD_REQ/CFM and CSR_IP_IFCONFIG_NAT_REQ/CFM signals have been added to control forwarding of IP datagrams and network address translation.

3.13 csr_ip_ether_*.h

The API has been changed in the following regard:

1. The flowControlResume function has been replaced by a new primitive:

   CSR_IP_ETHER_IF_FLOW_CONTROL_RESUME_REQ

2. The return value of the frameTxFunction no longer indicates flow control pause, it only indicates whether it was possible or not to transmit the frame. A flow control pause primitive has been added to replace the previous mechanism:

   CSR_IP_ETHER_IF_FLOW_CONTROL_PAUSE_REQ

3. In addition, the flow control state is now per individual priority level instead of global. An IP stack that does not support individual priority flow control is allowed to consider flow control on any priority level as

a global flow control state (similar to previous API), or disregard flow control entirely and drop frames unconditionally when the frameTxFunction indicates that a frame cannot be sent.

4. The CsrIpEtherContext type has been split into two distinct types, CsrIpEtherIfContext and CsrIpEtherIpContext in order to make it more clear that the context passed in the CSR_IP_ETHER_IF_ADD_REQ holds the value to be passed in the corresponding CsrIpEtherFrameTxFunction, and the context passed in the CSR_IP_ETHER_IF_ADD_CFM holds the value to be passed in the corresponding CsrIpEtherFrameRxFunction. Consequently the member/variable names have been renamed to ifContext for the former, and ipContext for the latter.

5. The parameters of all responses have changed:

   appHandle, result, ifHandle -> ifHandle, result

6. The CSR_IP_ETHER_IF_MULTICAST_ADDR_GET_RES and CSR_IP_ETHER_IF_MULTICAST_ADDR_GET_IND have been removed as they are unused.

Please see the API documentation for further information.

3.14 csr_ip_socket_*.h

The dataLength field of the CSR_IP_SOCKET_TCP_DATA_REQ has been changed to a CsrUint16 (was CsrUint32).

Added the following primitives for configuring multicast on UDP sockets:

```
CSR_IP_SOCKET_UDP_MULTICAST_SUBSCRIBE_REQ
CSR_IP_SOCKET_UDP_MULTICAST_UNSUBSCRIBE_REQ
CSR_IP_SOCKET_UDP_MULTICAST_INTERFACE_REQ
CSR_IP_SOCKET_UDP_MULTICAST_SUBSCRIBE_CFM
CSR_IP_SOCKET_UDP_MULTICAST_UNSUBSCRIBE_CFM
CSR_IP_SOCKET_UDP_MULTICAST_INTERFACE_CFM
```

Added CSR_IP_SOCKET_OPTIONS_REQ and CSR_IP_SOCKET_OPTIONS_CFM primitives for controlling socket options.

The CSR_IP_SOCKET_RESULT_INVALID_HANDLE result code was not used (by design) and has been removed.

3.15 csr_exceptionhandler.h

The following parameters are now qualified with const:

```
csr_exceptionhandler_t - <anonymous parameter 1>
CsrStateEventExceptionWithInfo - theTask
CsrGeneralExceptionWithInfo - theTask
CsrGeneralWarningWithInfo - theTask
```

3.16 Target Scripts

The CSR_LOGTRANSPORT_HAVE_*, BSP_SUPPORT_TRANSPORT_* and BSP_SUPPORT_FW_EXTENSIONS make variables have been removed from the target scripts, as they are no longer needed.

3.17 csr_usr2usr_socket_cli and csr_usr2usr_socket_srv

These libraries have been removed from the pclin BSP along with the interface header platform/csr_peer_com.h.

4 Document References

Document	Reference

Document History

Revision	Date	History
1	02 DEC 10	Ready for release 3.0.0
2	Aug 11	Ready for release 3.1.0

CSR Synergy Framework 3.1.0 API Changes

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.