X Session Management Library

X Consortium Standard

Ralph Mor, X Consortium

X Session Management Library: X Consortium Standard

© !

" ! # $ $ % % $ $ & # % # # %
! $ ' () $ & *+ #) $ & & % % # & !
% ! # $! # % #, % $) $ & # !
& !) $ & $ # # - % $ & % # .

/ 0 % % # ! % % # # % $
) $ &

/12)34/56 2 7) " 3 7828 (6) 7)* 57/139/ 56 6:/; 34 6:; <7:8 2 " 2)) 3 7 "=728 7: =987:>
?9/ :3/ =7 7/28 /3 /12 56 6:/72) 34 2 16:/6?7=7/; 47/:2)) 43 6 "6 /7 9=6 "9 "3)2 6:8
:3:7:4 7:>2 2:/ 7: :3 2 2:/)16== /12 3:)3 /79 ?2 =76?=2 43 6:; =67 86 6>2) 3
3/12 =76?7=7/; 512/12 7: 6: 6 /73: 34 3:/ 6 / /3 / 3 3/12 57)2 6 7)7:> 4 3 39/ 34
3 7: 3::2 /73: 57/1 /12)34/56 2 3 /12 9)2 3 3/12 826=7:>) 7: /12)34/56 2

2@% % # % ! $! # #0 &
! #) $ & & & A $! !

5 # &) ! # ! B $ / 3 >

Table of Contents

30 0 & $) !
C /) ! = C
9 # #) D 8 # % 7 2
1 # 4 # = : !
E) ! ')!%+ 4 % E
%) E
/) 0 ; $ %B
/ 8 %B H
/) 0 ! %B
/) # & % # %B
%
$ %B
) 8 # 0) ! "
7 % 5 9
F) 0 ; $ C
F) 0 ; $ " C
!) 0 ; $
9)!% 7 $! 4 %
2 1 #
G) !) 0 ')! + 4 % G
7 A = G
/ %B
/ 7 % F %B
/ 7 % 8 %B C
/) 0 ; $ F %B C
/) 0 ; $ " C F %B C
/) 0 ; $ 8 %B C
/ % # %B C
/) " %B CC
/ 8 " %B CC
/ > " %B CC
C
) #) 0 ; $ C
) #) 0 ; $ " C C
) # 7 % CE
) #) 0 ! CE
) # 8 CE
%) # & CE
" CE
" CG
9 6$ 8 % % CG
9)! 7 $! 4 % CG
2 1 # C
) ! " C
H 4 8 C
6 % $
5 B / # # 2 0 !
6%B & # ! E

Chapter 1. Overview of Session Management

/ $) ! " % ') "+ 0 # $! ! %
! $ 0 # 6 $ % % $
& % % / % # & B 0 % % #
/ ! % !! # % $ $ /
% !! # ! % % 0 ! 7 @ % # %
& 0 % & % % # ! # !
$ # 0 ! # 6 % D ! % # $!
$ % # # % $ &
$ $ % !! # % / ! & % % #
% $ # %

4 $ % $! # $ # % %
! 6 % % % % %
! % % # # 1 & 0 % ! % % # !
% % #

Chapter 2. The Session Management Library

/) ! = ') + & 0 I I $ %) " 7
@ % # 0 B % & # ! $ # $
! ! $! % 1 0 B ! # 0 # $!
% $$ % # &

) & .

J 3 $ $ % $ % & $

J 3 $ $ % $! %

) ! % & $ $ % # %
/ & ! # % $
6 @ ! ! ! % # @ # $ # @ $
! ! / ! ! $ # ! !
% ! #

A % % ! #
F # $! / ! !
$ $ % % / # % % D
0 ! # ! ! # % & % % # '0
)! !! #+ @ % # 0 % &
& ! $! 0 ! # # # 4
@ ! % ! % 78 $ $! & %
$ % / % 78 # 0 # $
)! !! # % & ! 78 $ #

3 % % A $ & ! ! #
! $! ! 4 @ ! ! B
0 ! 7 % $ # & ! !
0 % % % % % & # % % # &

Chapter 3. Understanding SMlib's Dependence on ICE

/) ! " % # $ 7 2@%
'7 2+ " % / 7 2 % # # ! @ 0 % 0
% % 6 & B &) F B & # $
& 7 2 & B

/ 7 2 A % %B % ! 5 % # %
7 2 % % # % IceProcessMessages
$ % IceProcessMessages & # ! # # B !-
% # # # ! & % % ! & # # $ /
% & # %B ! # #
$$ % 0 % %B

/ ! & $ %) ! 0 !
% # # % & # # 7 2 % % / % #
0 select % $ # % $ 7 2 % % ! %
XtAppAddInput & # % %B & 0 B IceProcessMes-
sages % ! # # 7 2 % %

/ $ % ! % B & & % $ # % % select
F # # $ & 7 2 % % % # 3 % #
% ! ! # SmcOpenConnection # % % &
! SmcOpenConnection & ! B % IceOpenCon-
nection & % & # ! $ 7 2 % % # @ &
% # ! B % % & # @ #
& 7 2 % % & % # / ! & $
% # 0 & 7 2 % % % # # #
% % % # & / # & 7 2
% & % % # & 0 B # % ! 7 2
% % # % # / & % % # % # # ##
! 0 7 2 $ # % $! $ # % % select

3 ! # & B # $$ / !
% ! % 0 % $ 7 2 % % / !
$ % IceListenForConnections # $ % %
$! % 3 % % % ! # % # IceAcceptConnection !
% # # ! % ! ## & 7 2 $ # %
$ # % % select

4 $ $! $ % # 7 2 % %
(7 2@% = * # #

Chapter 4. Header Files and Library Name

6 % ' ! # % + # % # # $
KX11/SM/SMlib.hL / # $ #$ $) # % #
$ % SMlib.h % # # $ K X11/SM/SM.hL & % #
$ $) %

?%) # # 72 % # B) #
72 (-lSM -lICE*

Chapter 5. Session Management Client (Smc) Functions

/ % # % &) ! % .

J %)

J % %

J #$ % %B

J) # # 0)

J 7 % &

J F ()0 ; $*

J F ()0 ; $" C*

J ! ()0 ; $*

J 9)!% $! $ %

J 1 # 2

Connecting to the Session Manager

/ % % & ! SmcOpenConnection

SmcConn **SmcOpenConnection**(*network_ids_list*, *context*, *xsmp_major_rev*, *xsmp_minor_rev*, *mask*, *callbacks*, *previous_id*, *client_id_ret*, *error_length*, *error_string_ret*);

network_ids_list) % $ & B 78' + $!

context 6 F -% NULL 9 # # ! $
7 2 % % % #' & MGN+

xsmp_major_rev / !- 0 $)" %

xsmp_minor_rev / ! 0 $)" %
'$ % $ # *xsmp_major_rev*+

mask 6 ! B # % & % % %B

callbacks / % %B / % %B # #
! $! !

previous_id / % 78 $! 0

client_id_ret / % 78 $ % #

error_length = $ *error_string_ret* ! #

error_string_ret ! # ! $ /
error_string_ret ! #!!
: ! *error_length* #

/ *network_ids_list* ! ! # % $
& B 78 $! # % !! 7$ *network_ids_list*
NULL 0 $ SESSION_MANAGER 0 ! 0 & # 2 %
& B 78 $ & $! .

tcp/<*hostname*>:<*portnumber*>

dec-
net/<*hostname*>::<*objname*>

local/<*hostname*>:<*path*> U

6 ! & !# $ & B 78 7$ $! &
!# % # & B 78 #

6$ % % # SmcOpenConnection % &
! 7$ % # $! 0
previous_id #% ! # % 78$!
0 7$ % $ - *previous_id* #
NULL 7$ *previous_id* % $ # # ! # 0 #
!) & % & *previous_id* NULL

7$ SmcOpenConnection %% # F % % $
SmcConn # *client_id_ret* ! % % 78 #$
/ *client_id_ret* # $ # & % free &
3 $ SmcOpenConnection NULL # $ $
error_string_ret

:) 72 % % % &
! 7$ 72 % % # @ & % # !
! $! 72 % % #$!
!

/ % @ ! #% && % 72 % %
& % 7$ % @ NULL % & &
% % 7$ % @ NULL % & 0
#72 % % #$$ NULL % @ % #&

6 0 # % #' % (9 # #) D 8 # % 72* +
% & 0 B %B $ & 72 % % % # # #
' IceAddConnectionWatch # IceRemoveConnectionWatch # & 0
% IceProcessMessages % ! select & # #
72 % % 4 $ $! (7 2@% = *
# #

/ % %B ! % $ % %B # # !
0 / ! B ! % $ & % % %B 6 $ %
%B % $ # 0 $) ! # / ! B ! %
! %B& # % ! $ 0 $

/ $ & 0 ! 3 # *mask* 0 .

```
SmcSaveYourselfProcMask
SmcDieProcMask
SmcSaveCompleteProcMask
SmcShutdownCancelledProcMask
```

4 % % %B % % % # 5)
0 B % %B & % #

```
typedef struct {

 struct {
  SmcSaveYourselfProc callback;
  SmPointer client_data;
 } save_yourself;

 struct {
  SmcDieProc callback;
  SmPointer client_data;
 } die;

 struct {
  SmcSaveCompleteProc callback;
  SmPointer client_data;
 } save_complete;

 struct {
  SmcShutdownCancelledProc callback;
  SmPointer client_data;
 } shutdown_cancelled;

} SmcCallbacks;
```

The Save Yourself Callback

/)0 ; $% %B $ SmcSaveYourselfProc

```
typedef void (*SaveYourselfProc)(smc_conn, client_data, save_type,
shutdown, interact_style, fast);
```

smc_conn / ! ! % % -%

client_data # %$#& % %B& #

save_type)%$ $ $! # 0#

shut_down)%$ $ #& B %

interact_style / $ % &#&

fast $/ % # 0 F %B

/ ! # ()0 ; $*! % % %B
- $! % 0 / % #
& A ! % SmcSetProperties # #%

& % 5 0 % %
SmcSaveYourselfDone

7$ *interact_style* SmInteractStyleNone % ! % &
& 0 7$ *interact_style* SmInteractStyleErrors %
! % & $ % # 7$ *interact_style*
SmInteractStyleAny % ! % & $
? % % % % & ! % ! %
SmcInteractRequest #& $ (7 % *! $! !
5 % # % & % SmcInteractDone /
% ! % SmcInteractRequest $ % 0 () 0 ; $*!
# $ % SmcSaveYourselfDone

7$ *save_type* SmSaveLocal % ! # $ %
%) % $ % # 0 $!
$ % 7 # $$ %
7$ *save_type* SmSaveGlobal & % % !! $ #
! %% 7$ *save_type* SmSaveBoth %
$ ' #$ % !! # ! $
+

) ! @ ! $ &.

J 7$ & # % & () 0 ; $* & $ SmSaveLocal
% # % ! $ % ## % % $ $
% $ % # % $ % # 7 & #
#)! !! # & $! $ #
! $

J 7$ & # % & () 0 ; $* & $ SmSaveGlobal
& # ! 0 % # #$

J 7$ & # % & () 0 ; $* & $ SmSaveBoth
& #$ 0 % # #$ 7 & # % ! $ &
$! % % $ % #& $ # #
4 & # #)! !! # & $!
$ # ! $

/ *shutdown* ! % $ & ! #& /
% #$$ # # & #& 7$
#& % # 0 #& $ () 0 ! *! 7$
#& % ! 0 # 0 %
% 0 (8 * () #& % #*

/ *fast* ! % $ % # 0 F %B
4 @ ! $! B & & $ & #
fast True

The Die Callback

/ 8 % %B $ SmcDieProc

```
typedef void (*SmcDieProc)(smc_conn, client_data);
```

smc_conn / ! ! % % - %

client_data # % $ # & % %B & #

/ ! # (8 * ! % & & #
/ % # # % SmcCloseConnection 6 !
0 & # () 0 ; $*! $ (8 * !

The Save Complete Callback

/)0 ! % %B $ SmcSaveCompleteProc

```
typedef void (*SmcSaveCompleteProc)(smc_conn, client_data);
```

smc_conn / ! ! % % - %

client_data # % $ # & % %B & #

The Shutdown Cancelled Callback

/) # & % # % %B $ SmcShutdownCancelledProc

```
typedef void (*SmcShutdownCancelledProc)(smc_conn, client_data);
```

smc_conn / ! ! % % - %

client_data # % $ # & % %B & #

/ ! # () # & % #*! & %
% # # & # % ' % E E (7 % 5
9 * + / % % & % $ # & # 0 # 7$
% % # SmcSaveYourselfDone % 0 #
% SmcSaveYourselfDone & %% ! False % %
& 0 # % SmcSaveYourselfDone & *success* !
$ % % ! $ 0

Closing the Connection

/ % % % & ! SmcCloseConnection

```
SmcCloseStatus SmcCloseConnection(smc_conn, count, reason_msgs);
```

smc_conn / ! ! % % - %

count / ! $ $ % % %

reason_msgs / $ % % %

/ *reason_msgs* ! & ! B NULL $ @ % #
% 3 & % $! # ! # / @
$! / ! ##
!

:) # 7 2 % % % &
! # 0 % ! ! ! % 0
7 2 % % 5 SmcCloseConnection % # 7 2 % % &
% # $ % 0 # & % % %B

7 2 # # $ IceAddConnectionWatch # IceRemoveConnectionWatch
& % %B 0 B # % ! 7 2 % % #
% # / % % %B ##, ! 0 7 2 $ # % $!
$ % 0 # % % select ' % XtAppAddInput XtRemoveInput+

SmcCloseConnection $ $ & 0 .

J SmcClosedNow 7 2 % % & % # ! & % % #
& 0 B # # % % & $ #

J SmcClosedASAP 73 # %% # % % SmcCloseCon-
nection % # & # IceProcessMessages / & % %
0 0 B # ! % % & $
'& 0 % A # IceProcessMessages
$ IceProcessMessagesConnectionClosed

J SmcConnectionInUse % % & % # ! %
% 0 %

Modifying Callbacks

/ ! # $ % %B SmcOpenConnection SmcModifyCallbacks

void **SmcModifyCallbacks**(*smc_conn*, *mask*, *callbacks*);

smc_conn / ! ! % % - %

mask 6 ! B # % & % % %B !# $

callbacks / & % %B

5 % $ 0 $ *mask* ! $ & 0 ! 3 #
.

SmcSaveYourselfProcMask
SmcDieProcMask
SmcSaveCompleteProcMask
SmcShutdownCancelledProcMask

Setting, Deleting, and Retrieving Session Management Properties

/ ! ! $ % SmcSetProperties

void **SmcSetProperties**(*smc_conn*, *num_props*, *props*);

smc_conn / ! ! % % - %

num_props / ! $

props / $

/ % $ # $ " 0
0 ! 0 & SmcSetProperties $ % :

! @ % # 0 &
? % $ % # !
$ % % $ %

4 # % $! ! #)!" %
% () ! " *

/ # 0 % SmcDeleteProperties

void **SmcDeleteProperties**(*smc_conn*, *num_props*, *prop_names*);

smc_conn / ! ! % % - %

num_props / ! $

prop_names / $

/ 0 # % SmcGetProperties

Status **SmcGetProperties**(*smc_conn*, *prop_reply_proc*, *client_data*);

smc_conn / ! ! % % - %

prop_reply_proc / % %B 0 B # & % !
%B

client_data / % # & # SmcPropRe-
plyProc % %B

/ 0 $ SmcGetProperties A $ $ # 0 0 $
%%

: # %B % ! %B
% %B $ SmcPropReplyProc 0 B # & # #

typedef void (***SmcPropReplyProc**)(*smc_conn*, *client_data*, *num_props*,
props);

smc_conn / ! ! % % - %

client_data / % # & # SmcPropReplyProc
% %B

num_props / ! $ #

props / $ #

/ $ % SmFreeProperty' % H (4 8 * + / $
% $ free

Interacting With the User

6$ % 0 () 0 ; $*! & *interact_style* $ SmInterac-
tStyleErrors SmInteractStyleAny % ! % % &
? % % % % & ! % ! %
SmcInteractRequest #& $ (7 % *! $! !

Status **SmcInteractRequest**(*smc_conn*, *dialog_type*, *interact_proc*, *client_data*);

smc_conn / ! ! % % -%

dialog_type / $# % &

interact_proc / % %B 0B#& (7 %*! 0
$! !

client_data / % # & # SmcInteractProc
% %B& (7 %*! 0

/ 0 $ SmcInteractRequest A $ $ # 0 0 $
%%

/ *dialog_type* ! %$ SmDialogError #%
% & # SmDialogNormal ! %
& #

: $ #& ! 0 $% %
#& 7$ #& % % # % 0 %#
& & % 0 () #& % #*! #$ (7 %*
!

/ SmcInteractProc % %B& 0B#& (7 %*! 0
$! !

typedef void (***SmcInteractProc**)(*smc_conn*, *client_data*);

smc_conn / ! ! % % -%

client_data # %$#& % %B& #

6$ % & ' (7 %*! + #
% SmcInteractDone

void **SmcInteractDone**(*smc_conn*, *cancel_shutdown*);

smc_conn / ! ! % % -%

cancel_shutdown 7$True #% F
#& % % #

/ *cancel_shutdown* ! ! True $ % # ()0
; $* %$# True$ #& # SmInteractStyleErrors SmInterac-
tStyleAny$ *interact_style*

Requesting a Save Yourself

/ F % %B $! ! SmcRequestSaveYourself

void **SmcRequestSaveYourself**(*smc_conn*, *save_type*, *shutdown*, *interact_style*, *fast*, *global*);

smc_conn / ! ! % % -%

save_type) % $ $ $! # 0 #

shutdown) % $ $ # & B %

interact_style / $ % & # &

fast 7$True % # 0 F %B

global & () 0 ; $ *

/ *save_type shutdown interact_style* # *fast* ! # % #
! # % E (/) 0 ; $ %B *

7$ *global* True () 0 ; $* #
% 4 @ ! 0 # $ 9 "&)
'9")+ ! % #) ! % & #! $
9") # $ # & $ &

7$ False () 0 ; $* # %
F #

Requesting a Save Yourself Phase 2

7 () 0 ; $* % ! F $!# &
% F % % 0 / # SmcRequestSaveYourselfPhase2

Status **SmcRequestSaveYourselfPhase2**(*smc_conn*, *save_yourself_phase2_proc*, *client_data*);

smc_conn / ! ! % % -%

save_type_phase2_proc / % %B 0 B # & () 0 ; $
" C*! 0 $! !

client_data / % # & #
SmcSaveYourselfPhase2Proc % %B & () 0
; $" C*! 0

/ 0 $ SmcRequestSaveYourselfPhase2 A $ $ #
0 0 $ %%

/ F ## % ! % '$ @ ! & # &
! & B % ! # + / ! ! !B
$ % ! # # %
0 #

Completing a Save Yourself

6$ 0 () 0 ; $* ! # % SmcSaveYourselfDone

void **SmcSaveYourselfDone**(*smc_conn*, *success*);

smc_conn / ! ! % % -%

success 7$True ()0 ; $* & %! # %% $

?$ % SmcSaveYourselfDone % ! 0 % F #
% % % #& !

Using Smc Informational Functions

```
int SmcProtocolVersion(smc_conn);
```

SmcProtocolVersion !- 0 $! ! %
% #&

```
int SmcProtocolRevision(smc_conn);
```

SmcProtocolRevision ! 0 $! !
% % #&

```
char *SmcVendor(smc_conn);
```

SmcVendor 0# ! # $% $ & $
! / # $ #& % free

```
char *SmcRelease(smc_conn);
```

SmcRelease 0# ! $!
/ # $ #& % free

```
char *SmcClientID(smc_conn);
```

SmcClientID ! # $ % 78 % #&
% % / $! & # SmcOpenConnection' 0#
# $ % 0 %+ free & % 78
##

```
IceConn SmcGetIceConnection(smc_conn);
```

SmcGetIceConnection 72% % -% % #&
! ! % % -% / 72% % -% % #
! ## $! % %)! $! $ $ %
& % % # 7% IceConnectionNumber IceConnec-
tionString IceLastSentSequenceNumber IceLastReceivedSequenceNumber #
IcePing 4 $ $! (7 2@% = * ##

Error Handling

7$ % % 0 @ % # % $! !
0B#) 6#$ # @ !
! stderr # @ $ 0 $ $ / %
% % # % SmcSetErrorHandler$ %

```
SmcErrorHandler SmcSetErrorHandler(handler);
```

/ # ; # NULL #$ #

SmcSetErrorHandler 0 #

/ SmcErrorHandler $ & .

```
typedef void (*SmcErrorHandler)(smc_conn, swap, offending_minor_opcode,
offending_sequence_num, error_class, severity, values);
```

smc_conn / ! ! % % -%

swap 6 $ #% $ %$ # 0 # &

offending_minor_opcode / ! %# $ $$ # !

offending_sequence_num / F % ! $ $$ # !

error_class / % $ $$ # !

severity IceCanContinue IceFatalToProtocol IceFatal-ToConnection

values 6 ## 0 %$% ! %# #%

: # 0B#$ % # / # 0B# & 73 %% IceSetIOErrorHandler 4 $ $! (7 2@% = * # #

Chapter 6. Session Management Server (Sms) Functions

/ % # % &) ! 0 .

J 7 A

J %

J) # ()0 ; $*!

J) # ()0 ; $" C*!

J) # (7 %*!

J) # ()0 ! *!

J) # (8 *!

J % #&

J

J " %

J $ % # % %

J 9)! $! $ %

J 1 #

Initializing the Library

SmsInitialize $) $ % # % # !
7 0# $! ! # % %B
& 0B# % ! &% % % !

Status **SmsInitialize**(*vendor*, *release*, *new_client_proc*, *manager_data*, *host_based_auth_proc*, *error_length*, *error_string_ret*);

vendor 6 %$! 0#

release 6 %$! !

new_client_proc %B 0B# % ! &% % %
!

manager_data 5 SmsNewClientProc % %B 0B#
! # & #

host_based_auth_proc 1 # % % %B

error_length = $ *error_string_ret* ! #

error_string_ret ! # ! $ /
error_string_ret # ! ! :
! *error_length* #

6$ SmsInitialize $ % % # ! # %
IceListenForConnections $ % $ & % % 6$ & # %
! % % % ! # % IceAcceptConnection

) % (6 % $ * $! # % '
% # # + 6 (7 2@% = *
# # $ $ # $ # %% 7 2 % %

2 % ! & % % % ! SmsNewClientProc
% %B 0 B # / ! & F % % - %
# $ $ % & % 6 !
! ! $ % %B # # $$!
% ! #

typedef Status (***SmsNewClientProc**)(*sms_conn*, *manager_data*, *mask_ret*, *callbacks_ret*, *failure_reason_ret*);

sms_conn 6 & F % % - %

manager_data # % $ # & % %B & #

mask_ret 3 # % & % % %B &
!

callbacks_ret 3 % % %B #
!

failure_reason_ret 4 #

7$ $ %% SmsNewClientProc # A &
% # $ *failure_reason_ret*) &
$ $!!

/ ! ! $ % %B # % 0 /
mask_ret ! % $ & % % %B 6 $ % %B % $ #
0 $) ! # / *mask_ret* ! % #
! %B& # % ! $ 0 $

/ $ & 0 ! 3 # ! B 0 .

SmsRegisterClientProcMask
SmsInteractRequestProcMask
SmsInteractDoneProcMask
SmsSaveYourselfRequestProcMask
SmsSaveYourselfP2RequestProcMask
SmsSaveYourselfDoneProcMask
SmsCloseConnectionProcMask
SmsSetPropertiesProcMask
SmsDeletePropertiesProcMask

SmsGetPropertiesProcMask

4 % % %B ! % ! #
% $ % % %B / & # % %B & 0 B #
)

```
typedef struct {
 struct {
  SmsRegisterClientProc callback;
  SmPointer manager_data;
 } register_client;

 struct {
  SmsInteractRequestProc callback;
  SmPointer manager_data;
 } interact_request;

 struct {
  SmsInteractDoneProc callback;
  SmPointer manager_data;
 } interact_done;

 struct {
  SmsSaveYourselfRequestProc callback;
  SmPointer manager_data;
 } save_yourself_request;

 struct {
  SmsSaveYourselfPhase2RequestProc callback;
  SmPointer manager_data;
 } save_yourself_phase2_request;

 struct {
  SmsSaveYourselfDoneProc callback;
  SmPointer manager_data;
 } save_yourself_done;

 struct {
  SmsCloseConnectionProc callback;
  SmPointer manager_data;
 } close_connection;

 struct {
  SmsSetPropertiesProc callback;
  SmPointer manager_data;
 } set_properties;

 struct {
  SmsDeletePropertiesProc callback;
  SmPointer manager_data;
 } delete_properties;

 struct {
```

```
  SmsGetPropertiesProc callback;
  SmPointer manager_data;
 } get_properties;

} SmsCallbacks;
```

The Register Client Callback

/ % %B $ % %B & 0 B # $ %
% % ! 7 SmsRegisterClientProc

typedef Status (***SmsRegisterClientProc**)(*sms_conn*, *manager_data*, *previous_id*);

sms_conn / ! ! % % - %

manager_data # % $ # & % %B & #

previous_id / % 78 $! 0

? $ $ % B % & % % !
& !

7$ % # $! 0 *previous_id* & %
! # % 78 $! 0
free *previous_id* & ## 7$ % $
- *previous_id* & NULL

7$*previous_id* 0 # ! # %
! / % %B # $ A & % & % !
% / % # & 0 O # NULL

3 & ! # % & F % 78
% SmsRegisterClientReply $ % ' # % # + #
SmsRegisterClientProc % %B # $

The Interact Request Callback

/ 7 % F % %B $ SmsInteractRequestProc

typedef void (***SmsInteractRequestProc**)(*sms_conn*, *manager_data*, *dialog_type*);

sms_conn / ! ! % % - %

manager_data # % $ # & % %B & #

dialog_type / $ # % &

5 % % 0 () 0 ; $*! & *interact_style* $ SmInteractStyleErrors SmInteractStyleAny % ! % % &
? % % % % & ! % !
F % & / ! # B F $
% & % 7 # # (7 % * ! %
! #& $ (7 % 8 *! $ % & @ %

/ *dialog_type* ! % $ SmDialogError # %
% & # SmDialogNormal ! %
& #

7$ # & ! 0 $ % %
& 7$ # & % % #' % $ # (7 % 8 *! +
! # # () # & % #*! % %
F # %

The Interact Done Callback

5 % # % & SmsInteractDoneProc %
%B & 0 B #

typedef void (***SmsInteractDoneProc**)(*sms_conn*, *manager_data*, *cancel_shutdown*);

sms_conn / ! ! % % - %

manager_data # % $ # & % %B & #

cancel_shutdown) % $ $ F # & %
% #

: # & % % % # $ % # () 0 ; $*
% $ # True $ # & # SmInteractStyleErrors SmInteractStyleAny
$ *interact_style*

The Save Yourself Request Callback

/) 0 ; $ F % %B $ SmsSaveYourselfRequestProc

typedef void (***SaveYourselfRequestProc**)(*sms_conn*, *manager_data*, *save_type*, *shutdown*, *interact_style*, *fast*, *global*);

sms_conn / ! ! % % - %

manager_data # % $ # & % %B & #

save_type) % $ $ $! # 0 #

shutdown) % $ $ # & B %

interact_style / $ % & # &

fast 7$ True % # 0 F %B

global & () 0 ; $*

/) 0 ; $ F ! ! % %B
& 4 $! *save_type shutdown interact_style* # *fast*
! % G () #) 0 ; $ *

7$ *global* True () 0 ; $* #
% 7$ *global* False () 0 ; $* #
% F #

The Save Yourself Phase 2 Request Callback

/) 0 ; $ " C F % %B $
SmsSaveYourselfPhase2RequestProc

typedef void (***SmsSaveYourselfPhase2RequestProc**)(*sms_conn*, *manager_data*);

sms_conn / ! ! % % - %

manager_data # % $ # & % %B & #

/ F % ! % '$ @ ! & # & !
& B % ! # +) % ! ! ! B
$ % ! # # %
0 #

The Save Yourself Done Callback

5 % # 0 () 0 ; $* !
SmsSaveYourselfDoneProc & 0 B #

typedef void (***SaveYourselfDoneProc**)(*sms_conn*, *manager_data*, *success*);

sms_conn / ! ! % % - %

manager_data # % $ # & % %B & #

success 7$True) 0 ; $ & % ! # %% $

? $ () 0 ; $ 8 * & % ! 0 % F #
% % # & !

The Connection Closed Callback

7$ % ! ' % SmcCloseConnection SmsCloseConnectionProc % %B 0 B #

typedef void (***SmsCloseConnectionProc**)(*sms_conn*, *manager_data*, *count*, *reason_msgs*);

sms_conn / ! ! % % - %

manager_data # % $ # & % %B & #

count / ! $!

reason_msgs / $ % % %

/ *reason_msgs* ! & ! B NULL # *count* ! A
' + $ @ % # 3 & % $!
! #/@ $! /
! ## !

SmFreeReasons $! 4 $ $! %
H (4 8 *

The Set Properties Callback

5 % ! ! SmsSetPropertiesProc
% %B & 0 B #

```
typedef void (*SmsSetPropertiesProc)(sms_conn, manager_data, num_props,
props);
```

sms_conn / ! ! % % - %

manager_data # % $ # & % %B & #

num_props / ! $

props / $

/ % $ # $ 4 # % $
! ! #)!" % % ()
! " *

" 0 0 ! 0 &) ! 0 #
$ # ! % / ! F # # $ #

/ $ % SmFreeProperty 4 $ $! % H
(4 8 * ; # $ % $ & % free

The Delete Properties Callback

5 % # ! ! SmsDeleteProper-
tiesProc % %B & 0 B #

```
typedef void (*SmsDeletePropertiesProc)(sms_conn, manager_data,
num_props, prop_names);
```

sms_conn / ! ! % % - %

manager_data # % $ # & % %B & #

num_props / ! $

prop_names / $ #

/ % $ # $ 4 # % $!
! #)!" % % () !
" *

The Get Properties Callback

/ SmsGetPropertiesProc % %B 0 B # & % & 0

```
typedef void (*SmsGetPropertiesProc)(sms_conn, manager_data);
```

sms_conn / ! ! % % - %

manager_data # % $ # & % %B & #

/ ! # # % SmsReturnProperties 6 $
$ % # #

Registering the Client

/ % ' SmsRegisterClientProc % %B+ SmsRegisterClientReply

```
Status SmsRegisterClientReply(sms_conn, client_id);
```

sms_conn / ! ! % % - %

client_id 6 ! # F % 78

/ 0 $ SmsRegisterClientReply A $ $ # 0 0
$ %% 4 & %% $) % % ! ! # % $
% 78 $ D & #

7$ NULL *previous_id* & % $ # & % # $
client_id # # % *previous_id*

3 & *client_id* # F 78 $ # !
7 ## ! # # () 0 ; $* ! &
type P Local *shutdown* P False *interact-style* P None # *fast* P False !
!# $ %

: % % 78 # % % B 78
$ 7$ % ! # # # & # !
78 7 # % 78 # $! ! % ! %
$! # $! ! ! &
$ % / % ! # & # $ % $ %
78 ! % 78 # F

; # % SmsGenerateClientID $ % F
% 78

```
char *SmsGenerateClientID(sms_conn);
```

sms_conn / ! ! % % - %

NULL & # $ 78 % # # 3 & 0
$ $ % % 78 7 # $ # & % free &
# #

Sending a Save Yourself Message

/ # () 0 ; $* % SmsSaveYourself

```
void SmsSaveYourself(sms_conn, save_type, shutdown, interact_style,
fast);
```

sms_conn / ! ! % % - %

save_type) % $ $ $! # 0 #

shutdown) % $ $ # & B %

interact_style / $ % & # &

fast 7$True % # 0 F %B

/ ! # () 0 ; $* ! % % %B
- $! % 0 / % #
& A ! () " * ! # # %
& % 5 0 % #
() 0 ; $ 8 * !

7$ *interact_style* SmInteractStyleNone % ! % &
& 0 7$ *interact_style* SmInteractStyleErrors %
! % & $ % # 7$ *interact_style*
SmInteractStyleAny % ! % & $
/ % ! # (7 % F * ! # & $ (7 % *
! $! ! $ % % & 5
% # % & # # (7 % 8 * !
/ (7 % F * ! % ! $ () 0 ; $* #
$ () 0 ; $ 8 *

7$ *save_type* SmSaveLocal % ! # $ %
%) % $ % # 0 $!
$ % 7 # $$ %
7$ *save_type* SmSaveGlobal & % % !! $ #
! %% 7$ *save_type* SmSaveBoth %
$ ' # $ % !! # ! $
+

/ *shutdown* ! % $ & # & /
% # $$ # # & # & 7$
# & % % 0 # ! ! 7$ # &
% ! 0 # ! 0 % % 0 (8 *
() # & % # * % # $ 0 &

/ *fast* ! % $ % # 0 F %B
4 @ ! $! B & & $ #
fast True

Sending a Save Yourself Phase 2 Message

7 # # () 0 ; $ " C* ! % SmsSaveYourselfPhase2

```
void SmsSaveYourselfPhase2(sms_conn);
```

sms_conn / ! ! % % - %

/ ! # ! % 0 () 0
; $ " C F * ! / ! $! %
% $ @ # # % % 0 % # &
%

Sending an Interact Message

/ # (7 % * ! % SmsInteract

void **SmsInteract**(*sms_conn*);

sms_conn / ! ! % % - %

/ (7 % * ! % 0 $ % &
5 % # % & ! # (7 % 8 *
! !

Sending a Save Complete Message

/ # () 0 ! * ! % SmsSaveComplete

void **SmsSaveComplete**(*sms_conn*);

sms_conn / ! ! % % - %

/ ! # ! & # & % %B /
% $ %

Sending a Die Message

/ # (8 * ! % SmsDie

void **SmsDie**(*sms_conn*);

sms_conn / ! ! % % - %

? $! ! # & $ (% #*
! $! % % (8 * ! !

Cancelling a Shutdown

/ % % # & SmsShutdownCancelled

void **SmsShutdownCancelled**(*sms_conn*);

sms_conn / ! ! % % - %

/ % % & % $ # & # 0 # 7 $ %
() 0 ; $ 8 * ! % 0 # #
() 0 ; $ 8 * & %% ! False % %
& 0 # # () 0 ; $ 8 * & *success* !
$ % % ! $ 0

Returning Properties

7 (> " * ! ! # % SmsReturnProperties

```
void SmsReturnProperties(sms_conn, num_props, props);
```

sms_conn / ! ! % % -%

num_props / ! $

props / $ %

/ # $ 4 # % $
! ! #)!" % % ()
! " *

Pinging a Client

/ % %B % 0 # IcePing $ % 0 # #
72 / # 72 % % ! # Sms-
GetIceConnection' % G C (9)! 7 $! 4 % * +

```
void IcePing(ice_conn, ping_reply_proc, client_data);
```

ice_conn 60 #72% % -%

ping_reply_proc / % %B 0B & " 0

client_data / & # IcePingReplyProc % %B

5 " # '$ 0 + IcePingReplyProc % %B &
0B# 6 ! # 0 ! $! # $ & %
! % @ % # # #

```
typedef void (*IcePingReplyProc)(ice_conn, client_data);
```

ice_conn 60 #72% % -%

client_data / % # %$ # % IcePing

Cleaning Up After a Client Disconnects

5 ! % 0 (% #* ! &
% % # % % #% SmsCleanUp $ %
$ % % -%

```
void SmsCleanUp(sms_conn);
```

sms_conn / ! ! % % -%

Using Sms Informational Functions

```
int SmsProtocolVersion(sms_conn);
```

SmsProtocolVersion !- 0 $! ! %
% #&

```
int SmsProtocolRevision(sms_conn);
```

SmsProtocolRevision ! 0 $! !
% % #&

char *SmsClientID(*sms_conn*);

SmsClientID ! # $ % 78 % #&
% % ; # % free & % 78
##

/ ! $ % SmsClientHostName / ! &
%

char *SmsClientHostName(*sms_conn*);

/ # $ $! *protocol/hostname* & *protocol* $
Qtcp decnet localR ; # % free # &
##

IceConn SmsGetIceConnection(*sms_conn*);

SmsGetIceConnection 72 % % - % % #&
! ! % % - % / 72 % % - % % #
! ## $! % %)! $! $ $ %
& % % # 7% IceConnectionNumber # IceLastSe-
quenceNumber 4 $ $! (7 2@% = *
# #

Error Handling

7$! % 0 @ % # % $! %
0B #) 6#$ # @ & % !
! ' # @ + / ! % % #
% SmsSetErrorHandler

SmsErrorHandler SmsSetErrorHandler(*handler*);

/ # ; # NULL #$ #

SmsSetErrorHandler 0 # / SmsErrorHandler
$ & .

typedef void (*SmsErrorHandler)(*sms_conn*, *swap*, *offending_minor_opcode*,
offending_sequence_num, *error_class*, *severity*, *values*);

sms_conn / ! ! % % - %

swap 6$ & % #% $ %$#0 #
&

offending_minor_opcode / ! %# $ $$ # !

offending_sequence_num / F % ! $ $$ # !

error_class / % $ $$ # !

severity IceCanContinue IceFatalToProtocol IceFatal-
ToConnection

values 6 ## 0 % $ % ! % #
%

: # 0 B # $ % # /
0 B # & 73 %% IceSetIOErrorHandler 4
$ $! (7 2@% = * # #

Chapter 7. Session Management Properties

2 % # $ #)!" % .

```
typedef struct {
 char *name; /* name of property */
 char *type; /* type of property */
 int num_vals; /* number of values */
 SmPropValue *vals; /* the list of values */
} SmProp;

typedef struct {
 int length; /* the length of the value */
 SmPointer value; /* the value */
} SmPropValue;
```

/) ! " % # $ $ # $ # 0
$ & % F # % / $ & % $
# $ # # # % & % F # 2 %
% # &

6 $)! 6 8H # % 0 6 $)!6
6;H # % $ 6 $)!=7)/ $6 6;H
% $ $

)! !! #	3) % $ %	)!=7)/ $6 6;H	;
)! 8 %	3) % $ %	)!6 6;H	:
)!8 % # !! #	3) % $ %	)!=7)/ $6 6;H	: S
)!2 0 !	3) % $ %	)!=7)/ $6 6;H	:
)!" % 78	3) % $ %	)!6 6;H	:
)!" !	3) % $ %	)!6 6;H	;
)! !! #	3) % $ %	)!=7)/ $6 6;H	;
)! !! #	3) % $ %	)!=7)/ $6 6;H	:
)!) 1	)! 6 8H	)! 6 8H	:
)!) # & ! ! #	3) % $ %	)!=7)/ $6 6;H	:
)!9 78	)!6 6;H	)!6 6;H	;

S F # $ # @ '$ @ ! $ +

J)! !! #

/ B)! !! # @% % $ %
/ # $$ % % # % 78
! 3 "3)7 ! # $)!=7)/ $6 6;H

J)! 8 %

3 "3)7 # ! % $ 0 $ % # %
)!" ! # # $)!6 6;H

J)!8 % # !! #

/ # % # % !! # % % !! # & # 0 #
% '# ! # $! % % + & % #
% # $! % 7$ % !! # % $ #
) & ! $ % D % # #
)! !! # 3 "3)7 ! #)!=7)/ $6 6;H

J)!2 0 !

3 "3)7 # ! & $)!=7)/ $6 6;H &
6 6;H & 0 ! 0 ! # 0 ! 0
0

J)!" % 78

/ % $ 3) % $ % # $ $ % 3 "3)7 !
% 0 $ getpid # = '# % ! +

J)!" !

/ ! $! 3 "3)7 ! #
$! # execve # # $)!6 6;H

J)! !! #

/ % !! # % % !! # & # 0 #
% '# ! # $! % % + & % %
% 3 "3)7 # ! $)!=7)/ $6
6;H # % $! ! argv
/ % !! # # % & %
$ # % 78

J)! !! #

6 %)!) 1 SmRestartAnyway
% $ % !! # # $$ % $ % # ! 0
0 # 6 @ ! % # & %
&))!) 1 SmRestartAnyway #
! / &) ' D F + #
& #)! !! # # $$ % $

J)!) 1

7$)!) 1 & % $
% $ 7$ % $ # SmRestartIfRunning !# /
0 $ & .

SmRestartIfRunning
SmRestartAnyway
SmRestartImmediately C
SmRestartNever

/ SmRestartIfRunning # % / % #
@ $ & # $ %

/ SmRestartAnyway #) %
@ 0 $ @ $ %
! # 7 # # #)
& $ & % % $ # # ! &
%

6 % SmRestartAnyway #)! !! # #
)!) # & !! # % !! # # $ %
$ @

/)! 7!! # B)! 6 & ##
% ! % 7$ % @) #
%

SmRestartNever % $ % # & #
@

J)!) # & !! #

/ % !! # @ % # # & ! % $ %
)!) 1 SmRestar-
tAnyway / % ! ! 0 0 # %
$ 6 @ ! % # % % !
! / % @ 6 # & & % !
$$ / % & #)!) 1 SmRestartAnyway
& #)!) # & !! # & # $$ % !

J)!9 78

) % $ 78 3 "3)7 # ! & % D !
' *pw_name* ! ! $ % &#+

Chapter 8. Freeing Data

/ $ #0# SmFreeProperty

void **SmFreeProperty**(*prop*);

prop / $

/ $ $! SmsCloseConnectionProc % %B SmFreeReasons

void **SmFreeReasons**(*count*, *reasons*);

count / ! $

reasons / $ $

Chapter 9. Authentication of Clients

6 # ! ! % # $726
% %% & 0 72 % .

J / $ & 72% % #

J / % # & " %) %% 72% %

/ % ! # 0 ! ! # # '
72 #) ! ! + 4 $ $
! (7 2@% = * # #

Chapter 10. Working in a Multi-Threaded Environment

/ # % ! # % &) '
$ 7 2 + # % IceInitThreads 4 $ $
! (7 2@% = * # #

Chapter 11. Acknowledgements

/ B $ & $ %) !
.T # ? & 2 8 0) > # T $$
) B ?)% $)& %B # B 5 @