
Juphoon Protocol Framework

XCAP Client

Published: Feb 2008

For more information on Juphoon Protocol Framework, see <http://www.juphoon.com>

Juphoon XCAP Client Function Definition

Juphoon System Software Corporation.

<http://www.juphoon.com>

Tel: +86-574-87304379

Fax: +86-574-87304379

Text Part Number:

Copyright © 2008, Juphoon System Software Corporation.

All rights reserved.

Contents

1. INTRODUCTION.....	4
1.1 JUPHOON XCAP CLIENT	4
1.2 AUDIENCE	4
1.3 SCOPE.....	4
1.4 ABBREVIATIONS	4
2. SYSTEM ENVIRONMENT.....	4
2.1 BASIC DATA TYPES	5
3. USER INTERFACES	6
3.1 STRUCTURES	6
3.1.1 <i>ST_XCAPC_REQ_MSG</i>	6
3.1.2 <i>PFN_XAUCPROCSTAT</i>	6
3.1.3 <i>ST_XCAPC_HOST</i>	6
3.1.4 <i>ST_XCAPC_URI</i>	7
3.1.5 <i>ST_ZOS_SSTR</i>	7
3.1.6 <i>ST_ZOS_DBUF</i>	7
3.2 UTILITY INTERFACES	7
3.2.1 <i>Xcapc_ReqMsgCreate</i>	7
3.2.2 <i>Xcapc_ReqMsgDelete</i>	8
3.2.3 <i>Xcapc_GetStatDesc</i>	8
3.3 USERFUL INTERFACES	9
3.3.1 <i>Xcapc_Open</i>	9
3.3.2 <i>Xcapc_Close</i>	9
3.3.3 <i>Xcapc_Put</i>	10
3.3.4 <i>Xcapc_Get</i>	10
3.3.5 <i>Xcapc_Del</i>	10
3.4 CONFIG INTERFACES.....	11
3.4.1 <i>Xcapc_CfgGetLogLevel</i>	11
3.4.2 <i>Xcapc_CfgGetCompactXml</i>	11
3.4.3 <i>Xcapc_CfgGetTaskPriority</i>	12
3.4.4 <i>Xcapc_CfgGetTaskStackSize</i>	12
3.4.5 <i>Xcapc_CfgGetTaskQueueSize</i>	12
3.4.6 <i>Xcapc_CfgGetTaskTimerNum</i>	13
3.4.7 <i>Xcapc_CfgGetConnTmrLen</i>	13
3.4.8 <i>Xcapc_CfgGetWaitRspTmrLen</i>	14
3.4.9 <i>Xcapc_CfgGetProductVal</i>	14
3.4.10 <i>Xcapc_CfgGetLocalIpv4</i>	14
3.4.11 <i>Xcapc_CfgGetLocalIpv6</i>	15
3.4.12 <i>Xcapc_CfgGetServIpv4</i>	15
3.4.13 <i>Xcapc_CfgGetServIpv6</i>	16
3.4.14 <i>Xcapc_CfgGetServHost</i>	16
3.4.15 <i>Xcapc_CfgSetLogLevel</i>	17

3.4.16	<i>Xcapc_CfgSetCompactXml</i>	17
3.4.17	<i>Xcapc_CfgSetTaskPriority</i>	17
3.4.18	<i>Xcapc_CfgSetTaskStackSize</i>	18
3.4.19	<i>Xcapc_CfgSetTaskQueueSize</i>	18
3.4.20	<i>Xcapc_CfgSetTaskTimerNum</i>	18
3.4.21	<i>Xcapc_CfgSetConnTmrLen</i>	19
3.4.22	<i>Xcapc_CfgSetWaitRspTmrLen</i>	19
3.4.23	<i>Xcapc_CfgSetProductVal</i>	20
3.4.24	<i>Xcapc_CfgSetLocalIpv4</i>	20
3.4.25	<i>Xcapc_CfgSetLocalIpv6</i>	20
3.4.26	<i>Xcapc_CfgSetServIpv4</i>	21
3.4.27	<i>Xcapc_CfgSetServIpv6</i>	21
3.4.28	<i>Xcapc_CfgSetServHost</i>	22
3.5	TASK INTERFACES.....	22
3.5.1	<i>Xcapc_Start</i>	22
3.5.2	<i>Xcapc_Stop</i>	23
3.6	VERSION INTERFACES.....	23
3.6.1	<i>Xcapc_GetVersion</i>	23
3.7	URI INTERFACES.....	23
3.7.1	<i>Xcapc_UriCreate</i>	23
3.7.2	<i>Xcapc_UriDelete</i>	24
3.7.3	<i>Xcapc_UriAddDocSeg</i>	24
3.7.4	<i>Xcapc_UriAddDocSegs</i>	25
3.7.5	<i>Xcapc_UriAddStepByName</i>	25
3.7.6	<i>Xcapc_UriAddStepByNameX</i>	26
3.7.7	<i>Xcapc_UriAddStepByPos</i>	26
3.7.8	<i>Xcapc_UriAddStepByPosX</i>	27
3.7.9	<i>Xcapc_UriAddStepByAttr</i>	28
3.7.10	<i>Xcapc_UriAddStepByAttrX</i>	28
3.7.11	<i>Xcapc_UriAddStepByPAttr</i>	29
3.7.12	<i>Xcapc_UriAddStepByPAttrX</i>	30
3.7.13	<i>Xcapc_UriAddQryByName</i>	31
3.7.14	<i>Xcapc_UriAddQryByNameX</i>	32
3.7.15	<i>Xcapc_UriSetTermsByAttr</i>	32
3.7.16	<i>Xcapc_UriSetTermsByAttrX</i>	33
3.7.17	<i>Xcapc_UriSetTermsByNs</i>	33
3.7.18	<i>Xcapc_UriFormatMsg</i>	34

List of Tables

TABLE 2-1: BASIC DATA TYPES.....	5
----------------------------------	---

List of Figures

FIGURE 2-1 XCAP CLIENT.....	5
-----------------------------	---

1. Introduction

1.1 Juphoon XCAP Client

The Extensible Markup Language (XML) Configuration Access Protocol (XCAP) is a set of conventions for mapping XML documents and document components into HTTP URIs, rules for how the modification of one resource affects another, data validation constraints, and authorization policies associated with access to those resources.

XCAP allows a client to read, write, and modify application configuration data stored in XML format on a server. Juphoon XCAP Client is used by user applications to perform XCAP client functions. Juphoon XCAP Client is strictly adhere to the standard and is compatible (can interoperate) with all existing software.

1.2 Audience

The readers of this document are assumed to have a working knowledge of the XCAP Client.

1.3 Scope

This document provides the definitions of interfaces provided by Juphoon XCAP Client to users. Details on the realization of the XCAP Client are not within the scope of this document.

1.4 Abbreviations

The following abbreviations are used in this document:

Abbreviation	Description
HTTP	Hypertext Transfer Protocol
URI	Uniform Resource Identifier
XCAP	The Extensible Markup Language (XML) Configuration Access Protocol

2. System Environment

This section describes the environment in which Juphoon XCAP Client is designed to operate. Figure 2-1 illustrates the framework.

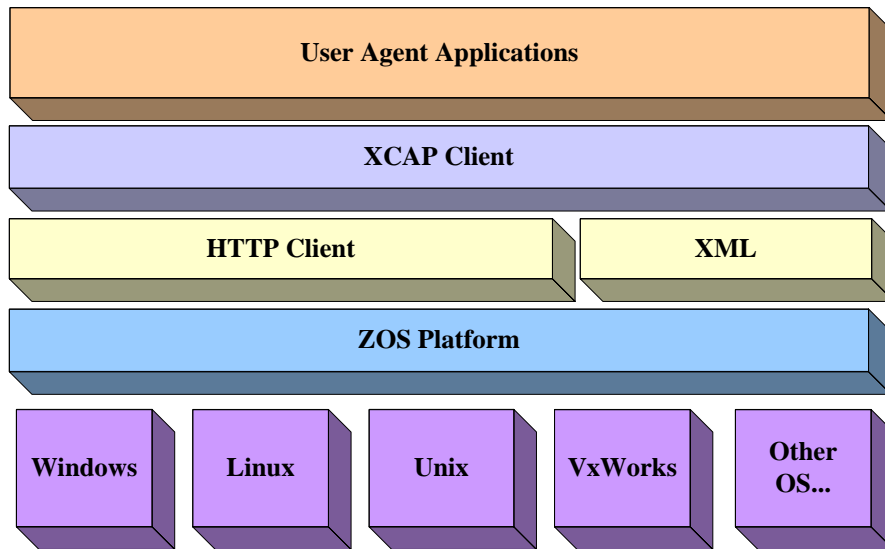


Figure 2-1 XCAP Client

2.1 Basic Data Types

There are some basic data types provided by ZOS platform. Table 2-1 lists these types used by Juphoon DNS Resolver.

Name	Type
ZDOUBLE	double
ZFLOAT	float
ZLONG	long
ZINT	int
ZSHORT	short
ZCHAR	char
ZULONG	unsigned long
ZUINT	unsigned int
ZSIZE_T	unsigned int
ZUSHORT	unsigned short
ZUCHAR	unsigned char
ZBOOL	int
ZVOID	void

Table 2-1: Basic Data Types

3. User Interfaces

3.1 Structures

3.1.1 ST_XCAPC_REQ_MSG

```
typedef struct tagXCAPC_REQ_MSG
{
    ZULONG dwMimeType;           /* mime type EN_XCAPC_MIME_TYPE */
    ST_XCAPC_URI *pstUri;        /* xcapc uri */
    ST_ZOS_SSTR stAuthName;     /* auth name */
    ST_ZOS_SSTR stAuthPasswd;   /* auth password */
    ST_ZOS_SSTR stIntendId;     /* 3gpp intended identity */
    ST_ZOS_SSTR stMatchEtag;    /* if-match entity tag */
    ST_ZOS_SSTR stNoMatchEtag; /* if-no-match entity tag */
    ST_XML_MSG *pstXmlMsg;      /* xml message */
    ST_ZOS_DBUF *pstMemBuf;     /* memory buffer */
} ST_XCAPC_REQ_MSG;
```

3.1.2 PFN_XAUCPROCSTAT

```
typedef ZINT (*PFN_XAUCPROCSTAT)(ZULONG dwCookie, ZULONG dwAucId, \
    ZUCHAR ucStatType, ST_XCAPC_RSP_MSG *pstRspMsg);
```

3.1.3 ST_XCAPC_HOST

```
typedef struct tagXCAPC_HOST
{
    ZUCHAR ucType;               /* host type EN_HTTP_HOST */
    ZUCHAR aucSpare[3];         /* for 32 bit alignment */
    union
    {
        ZCHAR acName[XCAPC_HOST_NAME_LEN]; /* hostname */
        ZULONG dwIpv4Addr;                 /* IPv4address */
        ZUCHAR aucIpv6Addr[ZINET_IPV6_ADDR_SIZE]; /* IPv6reference */
    } u;
} ST_XCAPC_HOST;
```

3.1.4 ST_XCAPC_URI

```
typedef struct tagXCAPC_URI
{
    ST_XCAPC_SEG_LST stDocSegs;    /* document segments */
    ST_XCAPC_NODES stNodes;       /* node selector */
    ST_XCAPC_QRY_LST stQryLst;    /* query list */
    ST_ZOS_DBUF *pstMemBuf;       /* memory buffer */
} ST_XCAPC_URI;
```

3.1.5 ST_ZOS_SSTR

```
typedef struct tagZOS_SSTR
{
    ZCHAR *pcStr;                 /* string address */
    ZUSHORT wLen;                 /* string length */
    ZUCHAR aucSpare[2];          /* for 32 bit alignment */
} ST_ZOS_SSTR;
```

3.1.6 ST_ZOS_DBUF

```
typedef struct tagZOS_DBUF
{
    struct tagZOS_DBUF *pstNext;  /* next data buffer */
    ZPOOLID zPoolId;              /* memory pool id for dbuf alloc */
    ZULONG dwBufLen;              /* buffer used length */
    ZULONG dwDftBlkSize;          /* default data block size in buffer */
    ZUCHAR ucBufType;             /* buffer mode ZDBUF_TYPE_BYTE... */
    ZUCHAR ucRefCnt;              /* buffer reference count */
    ZUCHAR aucSpare[2];           /* for 32 bit alignment */
#ifdef ZOS_SUPT_DUMP
    ZDUMPID zDumpId;              /* stack dump */
#endif
    ST_ZOS_DBUF_DATA *pstHead;    /* the first data block in buffer */
    ST_ZOS_DBUF_DATA *pstTail;    /* the last data block in buffer */
} ST_ZOS_DBUF;
```

3.2 Utility Interfaces

These interfaces are included in `xcapc_util.h`.

3.2.1 Xcapc_ReqMsgCreate

Creates an XCAP request message.

```
ZINT Xcapc_ReqMsgCreate(ST\_XCAPC\_REQ\_MSG **ppstReqMsg);
```

[Parameters]

Input parameters:

None.

Output parameters:

[ST_XCAPC_REQ_MSG](#) *ppstReqMsg

The newly created request message.

[Return value]

Returns ZOK on success, or ZFAILED on failure.

3.2.2 Xcapc_ReqMsgDelete

Deletes an XCAP request message.

```
ZINT Xcapc_ReqMsgDelete(ST\_XCAPC\_REQ\_MSG *pstReqMsg);
```

[Parameters]

Input parameters:

[ST_XCAPC_REQ_MSG](#) *pstReqMsg

The message to delete.

Output parameters:

None.

[Return value]

Returns ZOK on success, or ZFAILED on failure.

3.2.3 Xcapc_GetStatDesc

Retrieves the status description of a status type.

```
ZCHAR * Xcapc_GetStatDesc(ZUCHAR ucStatType);
```

[Parameters]

Input parameters:

ZUCHAR ucStatType

The status type.

Output parameters:

None.

[Return value]

Returns the status description.

3.3 Useful Interfaces

These interfaces are included in `xcapc_ui.h`.

3.3.1 Xcapc_Open

An application usage client opens a resource.

```
ZINT Xcapc_Open(ZULONG dwCookie, PFN\_XAUCPROCSTAT pfnProcStat,  
                ZULONG *pdwAucId);
```

[Parameters]

Input parameters:

ZULONG dwCookie
The user cookie.

[PFN_XAUCPROCSTAT](#) pfnProcStat
The function which processes XCAP states.

ZULONG *pdwAucId
The application usage client ID.

Output parameters:

None.

[Return value]

Returns ZOK on success, or ZFAILED on failure.

3.3.2 Xcapc_Close

An application usage client closes a resource.

```
ZINT Xcapc_Close(ZULONG dwAucId);
```

[Parameters]

Input parameters:

ZULONG dwAucId
The application usage client ID.

Output parameters:

None.

[Return value]

Returns ZOK on success, or ZFAILED on failure.

3.3.3 Xcapc_Put

Sends a PUT request for a document, element, or attribute.

```
ZINT Xcapc_Put(ZULONG dwAucId, ST\_XCAPC\_REQ\_MSG *pstReqMsg);
```

[Parameters]

Input parameters:

ZULONG dwAucId

The application usage client ID.

[ST_XCAPC_REQ_MSG](#) *pstReqMsg

The PUT request to send.

Output parameters:

None.

[Return value]

Returns ZOK on success, or ZFAILED on failure.

3.3.4 Xcapc_Get

Sends a GET request for a document, element, or attribute.

```
ZINT Xcapc_Get(ZULONG dwAucId, ST\_XCAPC\_REQ\_MSG *pstReqMsg);
```

[Parameters]

Input parameters:

ZULONG dwAucId

The application usage client ID.

[ST_XCAPC_REQ_MSG](#) *pstReqMsg

The GET request to send.

Output parameters:

None.

[Return value]

Returns ZOK on success, or ZFAILED on failure.

3.3.5 Xcapc_Del

Sends a DELETE request for a document, element, or attribute.

```
ZINT Xcapc_Del(ZULONG dwAucId, ST\_XCAPC\_REQ\_MSG *pstReqMsg);
```

[Parameters]

Input parameters:

ZULONG dwAucId
The application usage client ID.

[ST_XCAPC_REQ_MSG](#) *pstReqMsg
The DELETE request to send.

Output parameters:

None.

[Return value]

Returns ZOK on success, or ZFAILED on failure.

3.4 Config Interfaces

These interfaces are included in xcapc_cfg.h.

3.4.1 Xcapc_CfgGetLogLevel

Gets the log level. The default log level is ZLOG_LEVEL_ALL.

```
ZINT Xcapc_CfgGetLogLevel(ZULONG *pdwLevel);
```

[Parameters]

Input parameters:

None.

Output parameters:

ZULONG *pdwLevel
The log level.

[Return value]

Returns ZOK.

3.4.2 Xcapc_CfgGetCompactXml

Gets the compact XML Boolean value. The default value is ZFALSE.

```
ZINT Xcapc_CfgGetCompactXml(ZBOOL *pbCompact);
```

[Parameters]

Input parameters:

None.

Output parameters:

```
ZBOOL *pbCompact  
The compact XML Boolean value.
```

[Return value]

Returns ZOK.

3.4.3 Xcapc_CfgGetTaskPriority

Gets the task priority. The default priority is ZTASK_PRIORITY_NORMAL.

```
ZINT Xcapc_CfgGetTaskPriority(ZINT *piPriority);
```

[Parameters]

Input parameters:

None.

Output parameters:

```
ZINT *piPriority  
The task priority.
```

[Return value]

Returns ZOK.

3.4.4 Xcapc_CfgGetTaskStackSize

Gets the task stack size. The default size is 16K.

```
ZINT Xcapc_CfgGetTaskStackSize(ZULONG *pdwStackSize);
```

[Parameters]

Input parameters:

None.

Output parameters:

```
ZULONG *pdwStackSize  
The task stack size.
```

[Return value]

Returns ZOK.

3.4.5 Xcapc_CfgGetTaskQueueSize

Gets the task queue size. The default size is 100.

```
ZINT Xcapc_CfgGetTaskQueueSize(ZULONG *pdwQueueSize);
```

[Parameters]

Input parameters:

None.

Output parameters:

ZULONG *pdwQueueSize

The task queue size.

[Return value]

Returns ZOK.

3.4.6 Xcapc_CfgGetTaskTimerNum

Gets the number of task timers. The default number is 100.

```
ZINT Xcapc_CfgGetTaskTimerNum(ZULONG *pdwTimerNum);
```

[Parameters]

Input parameters:

None.

Output parameters:

ZULONG *pdwTimerNum

The number of task timers.

[Return value]

Returns ZOK.

3.4.7 Xcapc_CfgGetConnTmrLen

Gets the connect timer length. The default length is 5 seconds.

```
ZINT Xcapc_CfgGetConnTmrLen(ZULONG *pdwLen);
```

[Parameters]

Input parameters:

None.

Output parameters:

ZULONG *pdwLen

The connect timer length.

[Return value]

Returns ZOK.

3.4.8 Xcapc_CfgGetWaitRspTmrLen

Gets the time length for which a response will be waited. The default time length is 10 seconds.

```
ZINT Xcapc_CfgGetWaitRspTmrLen(ZULONG *pdwLen);
```

[Parameters]

Input parameters:

None.

Output parameters:

ZULONG *pdwLen

The time length.

[Return value]

Returns ZOK.

3.4.9 Xcapc_CfgGetProductVal

Gets the product value. The default value is "XDM-client/1.0 (Juphoon System)".

```
ZINT Xcapc_CfgGetProductVal(ZCHAR **ppcProductVal);
```

[Parameters]

Input parameters:

None.

Output parameters:

ZCHAR **ppcProductVal

The product value.

[Return value]

Returns ZOK.

3.4.10 Xcapc_CfgGetLocalIpv4

Gets the local IPv4 address. The default is "192.168.0.121".

```
ZINT Xcapc_CfgGetLocalIpv4(ZULONG *pdwIpv4, ZUSHORT *pwPort);
```

[Parameters]

Input parameters:

None.

Output parameters:

ZULONG *pdwIpv4
The IPv4 address.

ZUSHORT *pwPort
The port number.

[Return value]

Returns ZOK.

3.4.11 Xcapc_CfgGetLocalIpv6

Gets the local IPv6 address.

```
ZINT Xcapc_CfgGetLocalIpv6(ZUCHAR **ppucIpv6, ZUSHORT *pwPort);
```

[Parameters]

Input parameters:

None.

Output parameters:

ZUCHAR **ppucIpv6
The local IPv6 address.

ZUSHORT *pwPort
The port number.

[Return value]

Returns ZOK.

3.4.12 Xcapc_CfgGetServIpv4

Gets the server IPv4 address.

```
ZINT Xcapc_CfgGetServIpv4(ZULONG *pdwIpv4, ZUSHORT *pwPort);
```

[Parameters]

Input parameters:

None.

Output parameters:

ZULONG *pdwIpv4
The server IPv4 address.

ZUSHORT *pwPort
The port number.

[Return value]

Returns ZOK.

3.4.13 Xcapc_CfgGetServIpv6

Gets the server IPv6 address.

```
ZINT Xcapc_CfgGetServIpv6(ZUCHAR **ppucIpv6, ZUSHORT *pwPort);
```

[Parameters]

Input parameters:

None.

Output parameters:

ZUCHAR **ppucIpv6
The server IPv6 address.

ZUSHORT *pwPort
The port number.

[Return value]

Returns ZOK.

3.4.14 Xcapc_CfgGetServHost

Gets the server host.

```
ZINT Xcapc_CfgGetServHost(ST\_XCAPC\_HOST **ppstHost);
```

[Parameters]

Input parameters:

None.

Output parameters:

[ST_XCAPC_HOST](#) **ppstHost
The server host.

[Return value]

Returns ZOK.

3.4.15 Xcapc_CfgSetLogLevel

Sets the log level.

```
ZINT Xcapc_CfgSetLogLevel (ZULONG dwLevel);
```

[Parameters]

Input parameters:

ZULONG dwLevel
The log level.

Output parameters:

None.

[Return value]

Returns ZOK.

3.4.16 Xcapc_CfgSetCompactXml

Sets the compact XML Boolean value.

```
ZINT Xcapc_CfgSetCompactXml (ZBOOL bCompact);
```

[Parameters]

Input parameters:

ZBOOL bCompact
The compact XML Boolean value.

Output parameters:

None.

[Return value]

Returns ZOK.

3.4.17 Xcapc_CfgSetTaskPriority

Sets the task priority.

```
ZINT Xcapc_CfgSetTaskPriority (ZINT iPriority);
```

[Parameters]

Input parameters:

ZINT iPriority
The task priority.

Output parameters:

None.

[Return value]

Returns ZOK.

3.4.18 Xcapc_CfgSetTaskStackSize

Sets the task stack size.

```
ZINT Xcapc_CfgSetTaskStackSize(ZULONG dwStackSize);
```

[Parameters]**Input parameters:**

ZULONG dwStackSize
The task stack size.

Output parameters:

None.

[Return value]

Returns ZOK.

3.4.19 Xcapc_CfgSetTaskQueueSize

Sets the task queue size.

```
ZINT Xcapc_CfgSetTaskQueueSize(ZULONG dwQueueSize);
```

[Parameters]**Input parameters:**

ZULONG dwQueueSize
The task queue size.

Output parameters:

None.

[Return value]

Returns ZOK.

3.4.20 Xcapc_CfgSetTaskTimerNum

Sets the number of timers.

```
ZINT Xcapc_CfgSetTaskTimerNum(ZULONG dwTimerNum);
```

[Parameters]

Input parameters:

ZULONG dwTimerNum
The number of timers.

Output parameters:

None.

[Return value]

Returns ZOK.

3.4.21 Xcapc_CfgSetConnTmrLen

Sets the connect timer length by the second.

```
ZINT Xcapc_CfgSetConnTmrLen(ZULONG dwLen);
```

[Parameters]

Input parameters:

ZULONG dwLen
The connect timer length.

Output parameters:

None.

[Return value]

Returns ZOK.

3.4.22 Xcapc_CfgSetWaitRspTmrLen

Sets the time length for which a response will be waited. The time unit is second.

```
ZINT Xcapc_CfgSetWaitRspTmrLen(ZULONG dwLen);
```

[Parameters]

Input parameters:

ZULONG dwLen
The time length.

Output parameters:

None.

[Return value]

Returns ZOK.

3.4.23 Xcapc_CfgSetProductVal

Sets the product value.

```
ZINT Xcapc_CfgSetProductVal (ZCHAR *pcProductVal);
```

[Parameters]

Input parameters:

ZCHAR *pcProductVal

The product value.

Output parameters:

None.

[Return value]

Returns ZOK on success, or ZFAILED on failure.

3.4.24 Xcapc_CfgSetLocalIpv4

Sets the local IPv4 address.

```
ZINT Xcapc_CfgSetLocalIpv4 (ZULONG dwIpv4, ZUSHORT wPort);
```

[Parameters]

Input parameters:

ZULONG dwIpv4

The local IPv4 address.

ZUSHORT wPort

The port number.

Output parameters:

None.

[Return value]

Returns ZOK.

3.4.25 Xcapc_CfgSetLocalIpv6

Sets the local IPv6 address.

```
ZINT Xcapc_CfgSetLocalIpv6(ZUCHAR *pucIpv6, ZUSHORT wPort);
```

[Parameters]

Input parameters:

ZUCHAR *pucIpv6

The local IPv6 address.

ZUSHORT wPort

The port number.

Output parameters:

None.

[Return value]

Returns ZOK.

3.4.26 Xcapc_CfgSetServIpv4

Sets the server IPv4 address.

```
ZINT Xcapc_CfgSetServIpv4(ZULONG dwIpv4, ZUSHORT wPort);
```

[Parameters]

Input parameters:

ZULONG dwIpv4

The server IPv4 address.

ZUSHORT wPort

The port number.

Output parameters:

None.

[Return value]

Returns ZOK.

3.4.27 Xcapc_CfgSetServIpv6

Sets the server IPv6 address.

```
ZINT Xcapc_CfgSetServIpv6(ZUCHAR *pucIpv6, ZUSHORT wPort);
```

[Parameters]

Input parameters:

```
ZUCHAR *pucIpv6
The server IPv6 address.
```

```
ZUSHORT wPort
The port number.
```

Output parameters:

None.

[Return value]

Returns ZOK.

3.4.28 Xcapc_CfgSetServHost

Sets the server host.

```
ZINT Xcapc_CfgSetServHost (ST\_XCAPC\_HOST *pstHost);
```

[Parameters]**Input parameters:**

```
ST\_XCAPC\_HOST *pstHost
The server host.
```

Output parameters:

None.

[Return value]

Returns ZOK.

3.5 Task Interfaces

These interfaces are included in xcapc_task.h.

3.5.1 Xcapc_Start

Starts the XCAP Client task.

```
ZINT Xcapc_Start();
```

[Parameters]**Input parameters:**

None.

Output parameters:

None.

[Return value]

Returns ZOK.

3.5.2 Xcapc_Stop

Stops the XCAP Client task.

```
ZVOID Xcapc_Stop();
```

[Parameters]

Input parameters:

None.

Output parameters:

None.

[Return value]

None.

3.6 Version Interfaces

These interfaces are included in xcapc_version.h.

3.6.1 Xcapc_GetVersion

Returns the version of XCAP client.

```
ZCHAR * Xcapc_GetVersion();
```

[Parameters]

Input parameters:

None.

Output parameters:

None.

[Return value]

Returns the version of the XCAP client.

3.7 URI Interfaces

These interfaces are included in xcapc_uri.h.

3.7.1 Xcapc UriCreate

Creates an XCAP URI.

```
ZINT Xcapc UriCreate(ST XCAPC URI **ppstUri);
```

[Parameters]

Input parameters:

None.

Output parameters:

[ST_XCAPC_URI](#) **ppstUri

The newly created XCAP URI.

[Return value]

Returns ZOK on success, or ZFAILED on failure.

3.7.2 Xcapc UriDelete

Deletes an XCAP URI.

```
ZINT Xcapc UriDelete(ST\_XCAPC\_URI *pstUri);
```

[Parameters]

Input parameters:

[ST_XCAPC_URI](#) *pstUri

The XCAP URI to delete.

Output parameters:

None.

[Return value]

Returns ZOK on success, or ZFAILED on failure.

3.7.3 Xcapc UriAddDocSeg

Adds a document segment.

```
ZINT Xcapc UriAddDocSeg(ST\_XCAPC\_URI *pstUri, ST\_ZOS\_SSTR *pstSeg);
```

[Parameters]

Input parameters:

[ST_XCAPC_URI](#) *pstUri

The XCAP URI where the document segment is to be added.

[ST_ZOS_SSTR](#) *pstSeg

The document segment to add.

Output parameters:

None.

[Return value]

Returns ZOK on success, or ZFAILED on failure.

3.7.4 Xcap UriAddDocSegs

Adds document segments.

```
ZINT Xcap UriAddDocSegs (ST_XCAPC_URI *pstUri, ST_ZOS_SSTR *pstSegs);
```

[Parameters]

Input parameters:

[ST_XCAPC_URI](#) *pstUri

The XCAP URI where the document segments is to be added.

[ST_ZOS_SSTR](#) *pstSegs

The document segments to add.

Output parameters:

None.

[Return value]

Returns ZOK on success, or ZFAILED on failure.

3.7.5 Xcap UriAddStepByName

Adds a by-name step into an element selector in a URI.

```
ZINT Xcap UriAddStepByName (ST_XCAPC_URI *pstUri,
                             ST_ZOS_SSTR *pstPrefix, ST_ZOS_SSTR *pstName);
```

[Parameters]

Input parameters:

[ST_XCAPC_URI](#) *pstUri

The URI.

[ST_ZOS_SSTR](#) *pstPrefix

The prefix of the by-name.

[ST_ZOS_SSTR](#) *pstName

The name of the by-name.

Output parameters:

None.

[Return value]

Returns ZOK on success, or ZFAILED on failure.

3.7.6 Xcapc UriAddStepByNameX

Adds a step of by-name into an element.

```
ZINT Xcapc UriAddStepByNameX(ST XCAPC URI *pstUri,  
                             ST_ZOS_SSTR *pstPrefix, ZULONG dwNsId, ZULONG dwNameId);
```

[Parameters]

Input parameters:

[ST XCAPC URI](#) *pstUri

The URI.

[ST_ZOS_SSTR](#) *pstPrefix

The prefix of the by-name.

ZULONG dwNsId

The namespace ID.

ZULONG dwNameId

The name ID of the by-name.

Output parameters:

None.

[Return value]

Returns ZOK on success, or ZFAILED on failure.

3.7.7 Xcapc UriAddStepByPos

Adds a by-pos step into an element selector in a URI.

```
ZINT Xcapc UriAddStepByPos(ST XCAPC URI *pstUri,  
                           ST_ZOS_SSTR *pstPrefix, ST_ZOS_SSTR *pstName, ZULONG dwPos);
```

[Parameters]

Input parameters:

[ST_XCAPC_URI](#) *pstUri

The URI.

[ST_ZOS_SSTR](#) *pstPrefix

The prefix of the by-pos.

[ST_ZOS_SSTR](#) *pstName

The name of the by-pos.

ZULONG dwPos

The position.

Output parameters:

None.

[Return value]

Returns ZOK on success, or ZFAILED on failure.

3.7.8 Xcapc UriAddStepByPosX

Adds a step of by-pos into an element selector in a URI.

```
ZINT Xcapc UriAddStepByPosX(ST\_XCAPC\_URI *pstUri,  
                             ST\_ZOS\_SSTR *pstPrefix, ZULONG dwNsId, ZULONG dwNameId,  
                             ZULONG dwPos);
```

[Parameters]

Input parameters:

[ST_XCAPC_URI](#) *pstUri

The URI.

[ST_ZOS_SSTR](#) *pstPrefix

The prefix of the by-pos.

ZULONG dwNsId

The namespace ID.

ZULONG dwNameId

The name ID of the by-pos.

ZULONG dwPos

The position.

Output parameters:

None.

[Return value]

Returns ZOK on success, or ZFAILED on failure.

3.7.9 Xcapc UriAddStepByAttr

Adds a step of by-attr into an element selector.

```
ZINT Xcapc UriAddStepByAttr(ST\_XCAPC\_URI *pstUri,
                            ST\_ZOS\_SSTR *pstPrefix, ST\_ZOS\_SSTR *pstName,
                            ST\_ZOS\_SSTR *pstTName, ST\_ZOS\_SSTR *pstTVal);
```

[Parameters]

Input parameters:

[ST_XCAPC_URI](#) *pstUri

The URI.

[ST_ZOS_SSTR](#) *pstPrefix

The prefix of the by-attr.

[ST_ZOS_SSTR](#) *pstName

The name of the by-attr.

[ST_ZOS_SSTR](#) *pstTName

The att-name in the attr-test in the by-attr.

[ST_ZOS_SSTR](#) *pstTVal

The att-value in the by-attr.

Output parameters:

None.

[Return value]

Returns ZOK on success, or ZFAILED on failure.

3.7.10 Xcapc UriAddStepByAttrX

Adds a step of by-attr into an element selector.

```
ZINT Xcapc UriAddStepByAttrX(ST\_XCAPC\_URI *pstUri,
                              ST\_ZOS\_SSTR *pstPrefix, ZULONG dwNsId, ZULONG dwENameId,
                              ZULONG dwTNameId, ST\_ZOS\_SSTR *pstTVal);
```

[Parameters]

Input parameters:

[ST_XCAPC_URI](#) *pstUri

The URI.

[ST_ZOS_SSTR](#) *pstPrefix

The prefix of the element.

ZULONG dwNsId

The namespace ID.

ZULONG dwENameId

The by-attr name ID.

ZULONG dwTNameId

The att-name ID in the attr-test.

[ST_ZOS_SSTR](#) *pstTVal

The att-value in the attr-test.

Output parameters:

None.

[Return value]

Returns ZOK on success, or ZFAILED on failure.

3.7.11 Xcapc_UriAddStepByPAttr

Adds a step of by-pos-attr into an element selector.

```
ZINT Xcapc_UriAddStepByPAttr(ST\_XCAPC\_URI *pstUri,  
                             ST\_ZOS\_SSTR *pstPrefix, ST\_ZOS\_SSTR *pstName, ZULONG dwPos,  
                             ST\_ZOS\_SSTR *pstTName, ST\_ZOS\_SSTR *pstTVal);
```

[Parameters]

Input parameters:

[ST_XCAPC_URI](#) *pstUri

The URI.

[ST_ZOS_SSTR](#) *pstPrefix

The prefix of the by-pos-attr.

[ST_ZOS_SSTR](#) *pstName

The name of the by-pos-attr.

ZULONG dwPos

The position.

[ST_ZOS_SSTR](#) *pstTName

The att-name in the attr-test in the by-pos-attr.

[ST_ZOS_SSTR](#) *pstTVal

The att-value in the attr-test in the by-pos-attr.

Output parameters:

None.

[Return value]

Returns ZOK on success, or ZFAILED on failure.

3.7.12 Xcapc UriAddStepByPAttrX

Adds a step of by-pos-attr into an element selector.

```
ZINT Xcapc UriAddStepByPAttrX(ST\_XCAPC\_URI *pstUri,  
                             ST\_ZOS\_SSTR *pstPrefix, ZULONG dwNsId, ZULONG dwENameId,  
                             ZULONG dwPos, ZULONG dwTNameId, ST\_ZOS\_SSTR *pstTVal);
```

[Parameters]

Input parameters:

[ST_XCAPC_URI](#) *pstUri

The URI.

[ST_ZOS_SSTR](#) *pstPrefix

The prefix of the by-pos-attr.

ZULONG dwNsId

The namespace ID.

ZULONG dwENameId

The by-pos-attr name ID.

ZULONG dwPos

The position.

[ST_ZOS_SSTR](#) *pstTName

The att-name in the attr-test in the by-pos-attr.

[ST_ZOS_SSTR](#) *pstTVal

The att-value in the attr-test in the by-pos-attr.

Output parameters:

None.

[Return value]

Returns ZOK on success, or ZFAILED on failure.

3.7.13 Xcapc_UriAddQryByName

Adds a by-name into the query component in a URI.

```
ZINT Xcapc_UriAddQryByName(SST\_XCAPC\_URI *pstUri,
                           ST\_ZOS\_SSTR *pstNsName, ST\_ZOS\_SSTR *pstPrefix);
```

[Parameters]

Input parameters:

[ST_XCAPC_URI](#) *pstUri

The URI.

[ST_ZOS_SSTR](#) *pstNsName

The namespace.

[ST_ZOS_SSTR](#) *pstPrefix

The prefix of the by-name.

Output parameters:

None.

[Return value]

Returns ZOK on success, or ZFAILED on failure.

3.7.14 Xcapc_UriAddQryByNameX

Adds a by-name into the query component in a URI.

```
ZINT Xcapc_UriAddQryByNameX(ST_XCAPC_URI *pstUri, ZULONG dwNsId,
ST_ZOS_SSTR *pstPrefix);
```

[Parameters]**Input parameters:**

ST_XCAPC_URI *pstUri

The URI.

ZULONG dwNsId

The namespace ID.

ST_ZOS_SSTR *pstPrefix

The prefix of the by-name.

Output parameters:

None.

[Return value]

Returns ZOK on success, or ZFAILED on failure.

3.7.15 Xcapc_UriSetTermsByAttr

Sets a terminal selector by using an attribute-selector.

```
ZINT Xcapc_UriSetTermsByAttr(ST_XCAPC_URI *pstUri,
ST_ZOS_SSTR *pstName);
```

[Parameters]**Input parameters:**

ST_XCAPC_URI *pstUri

The URI.

ST_ZOS_SSTR *pstName

The attribute name.

Output parameters:

None.

[Return value]

Returns ZOK on success, or ZFAILED on failure.

3.7.16 Xcapc_UriSetTermsByAttrX

Sets a terminal selector by using an attribute-selector.

```
ZINT Xcapc_UriSetTermsByAttrX(ST_XCAPC_URI *pstUri,
                               ZULONG dwNsId, ZULONG dwNameId);
```

[Parameters]**Input parameters:**

ST_XCAPC_URI *pstUri

The URI.

ZULONG dwNsId

The namespace ID.

ZULONG dwNameId

The name ID of the attribute-selector.

Output parameters:

None.

[Return value]

Returns ZOK on success, or ZFAILED on failure.

3.7.17 Xcapc_UriSetTermsByNs

Sets a terminal selector by using a namespace-selector.

```
ZINT Xcapc_UriSetTermsByNs(ST_XCAPC_URI *pstUri);
```

[Parameters]**Input parameters:**

ST_XCAPC_URI *pstUri

The URI.

Output parameters:

None.

[Return value]

Returns ZOK on success, or ZFAILED on failure.

3.7.18 Xcapc_UriFormatMsg

Formats a URI into a message.

```
ZINT Xcapc_UriFormatMsg(ST\_XCAPC\_URI *pstUri, ST\_ZOS\_DBUF *pstMsgBuf);
```

[Parameters]

Input parameters:

[ST_XCAPC_URI](#) *pstUri

The URI.

[ST_ZOS_DBUF](#) *pstMsgBuf

The message buffer.

Output parameters:

None.

[Return value]

Returns ZOK on success, or ZFAILED on failure.