

WISE Send Data

Description

The communication between two WISE devices is divided into two possibilities

- **Send / receive** a number
- **Send / receive** a data buffer

When number is send or received the data size is always 5 bytes - the first byte is a control byte (always 'L') and the remaining 4 bytes represents a 32 bit integer in a [little endian](#) format.

The data is sent with 512 byte segments. After each segment 4 byte response is received.

Maemo Platform

```
#define WISE_SEND_BUFFER_SIZE 512

int wise_send_buffer(int socket, const char* data, int data_size)
{
    int transmitted=0, total_sent=0, buffer_size=0;
    const char* ptr = data;
    char buf[10];

    if ( data==NULL || logger==NULL ) return WISE_CONNECTION_ERROR;

    /* send data size */

    if ( wise_send_int(socket, data_size) != WISE_OK ) return WISE_CONNECTION_ERROR;

    /* initial buffer size */
    if ( data_size < WISE_SEND_BUFFER_SIZE ) buffer_size = data_size;
    else buffer_size = WISE_SEND_BUFFER_SIZE;

    /* send data */
    while ( total_sent < data_size )
    {
        transmitted = write(socket, ptr, buffer_size);
        if ( transmitted < buffer_size ) return WISE_CONNECTION_ERROR;

        total_sent += transmitted;
        ptr += transmitted;

        if ( total_sent + WISE_SEND_BUFFER_SIZE <= data_size )
        {
            buffer_size = WISE_SEND_BUFFER_SIZE;
        }
        else buffer_size = data_size - total_sent;

        transmitted = read(socket, buf, 4);
        if ( transmitted < 4 || strcmp(buf, "ACK") != 0 )
            return WISE_CONNECTION_ERROR;
    }
    return WISE_OK;
}
```

}

S60 Platform

```
#define KwiseSendBufferSize 512

void CWISEBase::SendBufferL(RSocket& aSocket, const TDesc8& aData)
{
    TRequestStatus status;

    TInt bufferSize(0), totalSent(0), dataSize(aData.Length());
    TBuf8<4> buffer;
    TSockXfrLength len;

    // send data size
```

```
    SendIntL(aSocket, dataSize);
```

```
    // initial buffer size
    if ( dataSize < KwiseSendBufferSize ) bufferSize = dataSize;
    else bufferSize = KwiseSendBufferSize;

    // send data
    while ( totalSent < dataSize )
    {
        TPtrC8 ptr(aData.Mid(totalSent, bufferSize));

        aSocket.Write(ptr, status);
        User::WaitForRequest(status);
        User::LeaveIfError(status.Int());

        totalSent += bufferSize;
        if ( totalSent + KwiseSendBufferSize <= dataSize )
        {
            bufferSize = KwiseSendBufferSize;
        }
        else bufferSize = dataSize - totalSent;

        aSocket.RecvOneOrMore(buffer, 0, status, len);
        User::WaitForRequest(status);
        User::LeaveIfError(status.Int());
    }
}
```

Links

- [Archived:Wireless Information Sharing Engine](#)
- [WISE Protocol](#)

