No. 13: The Web Connect API

World Telecom Labs has developed a method for external control of the IPNx switch. We have called this **Web Connect**. This can allow web-based applications to have access to a wide range of telephony functions. It also applies to PC-based programs too. This means it is now possible for customers quickly and easily to create their own value-added services to gain a market advantage. This API is designed to hide the complexity of call control procedures by giving a transactional interface to the client application.

The following case studies give some real world examples of how this API has been used:

Case Study 1 – Advertising Calls

An operator wishes to make phone calls to a predefined set of subscriber numbers. Those that answer will be played a pre-recorded advertising message from a list of messages stored on the IPNX. A report is produced on the success or failure of these calls. This solution was rapidly created by a 3rd party developer using a PC program to control the IPNX via Web Connect.

Case Study 2 – Web Enabled Calling (1)

A company with a web portal wanted to allow web browsers to get access to low cost long distance calling (not internet calls). This was achieved by using a java applet which collected the destination (B-leg) number from the user and then passed it to the IPNX to initiate the call. The IPNX was then instructed by the applet to call the user's number (the A-leg). Finally, the process was completed by the IPNX connecting the two legs of the call together.

For administration and billing purposes each user may be given an identifying code as their call is passed to the IPNX. This is included in the CDR that is created by the IPNX for that call. When the call completes the CDR is returned to the applet as well as being held by the IPNX.

Case Study 3 – Web Enabled Calling (2) – Directory Services

A special case of Case Study 2 involves a multi-national company who keep their internal telephone directory on a web page. Clicking on the desired extension establishes the call via the IPNX as described in the previous example.

Case Study 4 – Web Enabled Calling (3) – Assisted Browsing

Another variation on the above two Case Studies is the use of Click To Talk buttons on commercial web sites. When Click to Talk is used a call is made by the IPNX under Web Connect control to a Customer Agent at the company office. The user is prompted for the number they wish to be called back on and when this leg of the call has been successfully set up by the IPNX, as before the two parts of the call are bridged.

Web Connect Features Version 1.4

Communication takes place via a TCP socket. This ensures highly reliable message transport. The ***Web Connect*** API is message based with a Send and Acknowledge structure. The commands may include the following:

reset *client → WTL*

Used to reset the WTL switch, all ongoing calls created by the client application are cleared.

reset_ack *WTL → client*

Acknowledges a ***reset*** primitives. The client must wait for this primitive before sending an ***initiate*** primitive

initiate *client → WTL*

Requests the establishment of the two calls by the WTL switch. The client application can specify certain limitations to the call (for example, Maximum Call Duration, Maximum call progress time). Also the messages to be played to the user may be specified (for example, on A-leg while the B-leg is progressing and on A-leg before releasing the call if B-leg fails).

The IPNX will automatically establish, bridge and release the calls without any intervention from the client application. Calls established in this way are automatically given a special identifier to allow them to have an appropriate charging rate applied to them.

Three different telephone number field formats are supported by this API : a) E164 telephone number (international telephone number), b) IP address in dotted notation using NOP protocol, c) IP address in dotted notation using H323 protocol.

confirm *WTL → client*

Sent by the WTL switch to confirm that the initiate primitive is accepted and being processed.

reject *WTL → client*

The WTL switch sends this if an initiate is rejected. The cause field gives a reason for failure. The Cause value will be according to ISDN standard (see ETS 300 for list of cause value) with additional printable diagnostic message.

status **WTL → client**

Status report on the progression of the calls. For example, A-leg connected, playing message to end user or A-leg and B-leg connected, conversation starts. For a Pre-Paid call includes maximum duration before call will be cut-off.

release *WTL → client*

Both legs are released, the cause is reported as well as the CDR of both calls with the cost as calculated by the switch.

Note: The CDR format is:
duration in seconds, conversation start time as HH:MM:SS, start date as DD/MM/YY[YY], destination telephone number in international format, carrier number, cost of the call with 2 decimals (ex: 7.50), company id, balance left at the end of the call in balance unit

transfer *WTL → client*

Sent by the WTL switch to confirm that an initiate primitive containing a trigger is accepted and being transfer to the callback application.

reject *WTL → client*

The WTL switch sends this primitive if an initiate primitive is rejected. The cause field gives a reason for failure.

echo *client → WTL*

This switch responds to this primitive by sending an echo_ack primitive. It has non effect on the calls. The purpose of this primitive is to keep the TCP connection alive, if necessary.

echo_ack *WTL → client*

The switch sends this primitive in response to an echo primitive.

drop *client → WTL*

The client can use this to drop a call at any time. The WTL switch will respond with a release primitive or a reject primitive if the Call id is invalid.