TelcoBridges

BEST TECHNOLOGY > BRIGHT IDEAS > CLEAR FUTURE

Can You Build a Media Gateway in a Day?

YES! - Using TelcoBridges' *T*oolpack API

TelcoBridges is simplifying and accelerating telecom application development for developers building solutions for VoIP and TDM networks.

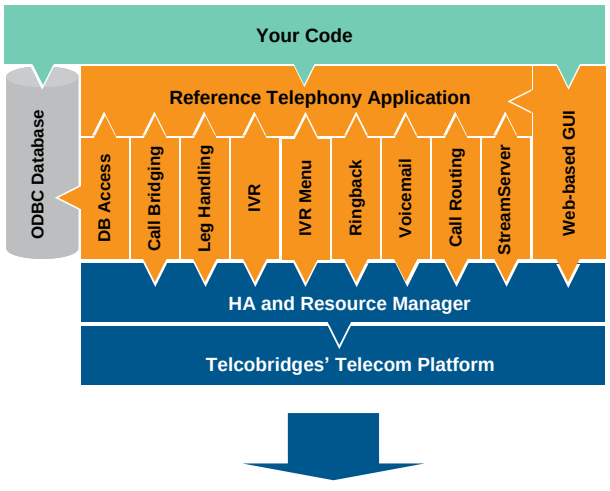
We Provide:		*You Get:*
Ready-to-use C++ tools	>	**Fast time-to-market**
The industry's best architecture	>	**A competitive advantage**
Source code availability	>	**The flexibility to define your future**

TelcoBridges' *T*oolpack is a set of pre-developed building blocks which are designed to dramatically shorten the development cycle while transparently incorporating all the benefits of TelcoBridges' carrier-grade architecture.

*TelcoBridges' T*oolpack API

- > 10 main function building blocks
- > Ready-to-compile C++ classes
- > Open: source code is provided
- > Independently used
- > Completely customizable
- > Operating system agnostic

Create your solution using the industry's best telecom platform and pre-developed application building blocks:



TelcoBridges' *T*oolpack provides customizable functions:

- Reference telephony application
  - Ready-to-use media gateway application
- Web-based configuration GUI
- Call bridging (VoIP, TDM, hair-pinning)
- Call leg handling
- IVR
- IVR menu
- Ringback tone
- Voicemail
- Call routing
- ODBC database access
- TB-StreamServer play/record audio files

The HA and Resource Manager transparently handles TelcoBridges' Telecom platform which includes:

- SIP, SS7, and ISDN signaling
- VoIP with a complete set of codecs
- T1/E1/J1, DS3, OC3/STM-1
- IVR (play, record, tone det/gen, conferencing
- Non-blocking scalability up to 32,768 channels
- Hardware and software redundancy

Using TelcoBridges' *T*oolpack development tools, the creation of a media gateway solution is simplified to the point where it can be achieved ***in a single day***.

Step 1: Define a new ***CSimpleCall*** class to inherit from TelcoBridges' ITBCMCLib (main library class)

```
class CSimpleCall : public ITBCMCLib
```

Step 2: Initialize the libraries in the ***Init*** method of ***CSimpleCall***

Step 3: In the ***OnCallLegPresent*** notification, which indicates an incoming call, allocate a call leg by creating a new CTBCMCCallLeg and supplying it the message handle

```
TBX_VOID CSimpleCall::OnCallLegPresent (   IN      PTBCMC_MSG_NOTIF_CALL_LEG_PRESENT      in_pMsg,
                                           IN      TBX_MSG_HANDLE                         in_hMsg
                                       )
{
  PCTBCAFCall                       pCall;
  TBCMC_LEG_ID                      LegId;
   PTBCMC_CALL_LEG_ATTRIBUTE        pAttribute;
   PCTBCMC_CALL_LEG_ATTRIBUTE       pIncomingLegAttribute;
   PCTBCMC_CALL_LEG_ATTRIBUTE       pOutgoingLegAttribute;
   ...

   // Extract information from message
   CTBCMCCallLeg::GetLegMsgInfo( in_hMsg, &LegId, &pAttribute );

   // Get the account for the call
   ...

   // Create an incoming call leg attribute object
   pIncomingLegAttribute = tbnew CTBCMC_CALL_LEG_ATTRIBUTE( in_hMsg, pAttribute );

   // Copy the extracted call leg attribute to the incoming call leg attribute object
   pOutgoingLegAttribute->CopyFrom( pAttribute );

   // Create an outgoing call leg attribute object
   pOutgoingLegAttribute = tbnew CTBCMC_CALL_LEG_ATTRIBUTE();

   // Copy the extracted call leg attribute to the outgoing call leg attribute object
   pOutgoingLegAttribute->CopyFrom( pAttribute );

   // Change the NAP (Network Access Point) of the outgoing call leg attribute from the routing information
      // First find the route for the call
      //    If we found the route
      //       Set the NAP
   ...

   // Create the call (i.e. which contains 2 call legs)
   pCall = tbnew CTBCAFCall( un32AccountId );

   // Add legs information
   pCall->AddIncoming( LegId, pIncomingLegAttribute );
   pCall->AddOutgoing( pOutgoingLegAttribute );

   // Bridge the two call legs
   pCall->InitCall();
   ...
```

Get your accounting information here based on the incoming call leg info.

Get your routing information here and assign an outgoing NAP (i.e. route, codec, etc.) to the call leg.

Step 4: **Watch the application bridge VoIP and TDM calls!**

What have you done today?

HEAD OFFICE
115-A1 De Vaudreuil,
Boucherville, Quebec,
Canada, J4B 1K7
T +1 450 655 8993
F +1 450 655 9511
info@telcobridges.com
www.telcobridges.com

ASIA
T +852 9033 0036 (HK)
T +86 139 2525 1005 (China)
F +852 3012 2782 (HK)

KOREA
T +82 2 564 0775
M +82 11 9741 5882
F + 82 2 3288 1302

THE AMERICAS
M +1 408-529 0451 (USA)
T +1 408 374 1004 (USA)
F +1 408 374 1591 (USA)

ABOUT US

TelcoBridges is clearly defining the future of enabling communications technologies. By supplying the industry's best telecom platform, **TelcoB**ridges is helping system integrators worldwide realize their bright ideas. Since 2002, **TelcoB**ridges' customers create carrier-grade telecom solutions used by the world's largest operators in more than 30 countries.

Finalist "2006 Canada Innovation Award": Development of Export Sales