

# Envox CT ADE™ V8.4 Release Notes

April 18, 2005

## System Requirements

---

### Operating System Support

CT ADE 8.4 supports Windows 2000, Windows XP, and Windows 2003.

### System Release Support

CT ADE 8.4 supports Intel Dialogic System Release 5.1.1 FP 1 or 6.0.

### Host Media Processing (HMP) Software Support

CT ADE 8.4 supports Intel\* NetStructure\* Host Media Processing Software 1.1.

### Supported ASR Engines

Microsoft SAPI 5.0 or higher.

Nuance Speech Recognition System 7.0.4 with Nuance Foundation Speech Objects 1.0a GA or later.

ScanSoft SpeechPearl\* 8.0 or higher.

SpeechWorks 6.5.

### Supported TTS Engines

Microsoft SAPI 5.0 or higher.

Nuance Vocalizer 1.0

Nuance Speech Recognition System 7.0.4 with Nuance Foundation Speech Objects 1.0a GA or later.

ScanSoft RealSpeak\* TTS engine version 3.5.1 or higher.

## What's New in Version 8.4?

---

This version of CT ADE supports Intel NetStructure Host Media Processing (HMP) Software version 1.1 without requiring special configuration steps.

Global Call initialization is now supported.

New SIP features are now supported, including: Message Waiting Indicator (MWI), call diversion information, blind transfer, supervised transfer, call hold/retrieve, and T.38 fax.

## Recent Fixes

The following problems have been fixed since the release of CT ADE Version 8.3.

---

**Problem:** Even though a GCEV\_UNBLOCKED event is in the logs, the Trunk Resource is taken to the NetworkDown state instead of the Idle state.

**Solution:** There was a race condition in the timer code responsible for setting the Trunk Resource to the NetworkDown state when it is first opened. Normally, if an Unblocked event is not seen within a certain amount of time, the Trunk Resource is set to the NetworkDown state. When the Unblocked event comes in, the Trunk Resource is set to the Idle state and the original timer is disabled. The race condition caused the Trunk to be set to NetworkDown regardless of the presence of an Unblocked event. This has been corrected.

**Problem:** Using REGID\_SwVrLoadForm to repeatedly load the same form into the SWVR Resource will cause a resource leak.

**Solution:** Load the form only once per VR Resource. If using ADL, do not load it again when a task is restarted. If using ADX, do not load it again if the assigned VR resource has not been changed.

*In ADL:*

The best way to design an ADL application that loads a form is to write a main loop in your task instead of using the "restart" statement. The form can be loaded once above the main loop to ensure it is only done one time per VR Resource.

*In ADX:*

Make sure the form is not being loaded in any places that will be repeated. For example, do not load the form in the IncomingCall event procedure. In Visual Basic, for example, it would be best to load the VR form when the Visual Basic Form is loading.

**Problem:** CT ADE was reporting false positive battery reversal.

**Solution:** This has been corrected. Please refer to the documentation on battery reversal.

**Problem:** If a task is in the TrunkWaitDisconnect function, and another task calls task\_hangup targeting the first task, nothing happens. The first task remains stuck in the TrunkWaitDisconnect function.

**Solution:** A software correction has been made so that TrunkWaitDisconnect now responds to task\_hangup. Previously it only responded to the Trunk's state changing to RemoteDisconnected. With the fix, the task will now wake up and jump to the onhangup handler as expected.

**Problem:** Setting constraints for recognition in a SpeechWorks module is normally done dynamically using the AddVocabulary API function. This API function was not exposed in a useful way from within CT ADE, so constraints were not supported.

**Solution:** A new REGID has been added to CT ADE's SwVr technology. REGID\_SwVrGrammarVocabType (2829) is now used to control the VocabType that is passed in to the ALTdmAddVocabulary API function when it is called by PlayAndRecogGrammar.

Constraints can be set by loading the vocab file through REGID\_SwVrLoadVocab (2817), setting the VocabType to "entry\_list" using REGID\_SwVrGrammarVocabType, and calling VrPlayAndRecogGrammar with that vocab. Wordspotting should not be used.

**This is the list of legal strings that can be used with REGID\_SwVrGrammarVocabType:**

ALTdm_COMMAND_VOCABULARY:	"command_vocabulary"
ALTdm_CREDITCARD_PREFIXES:	"creditcard_prefixes"
ALTdm_CURRENCY_ITEMS:	"currency_items"
ALTdm_DATE_ITEMS:	"date_items"
ALTdm_DTMF_COMMAND_VOCABULARY:	"dtmf_command_vocabulary"
ALTdm_DTMF_LIST_ITEMS:	"dtmf_list_items"
ALTdm_ENTRY_LIST:	"entry_list"
ALTdm_LIST_ITEMS:	"list_items"
ALTdm_NATURALNUMBERS_ITEMS:	"naturalnumbers_items"
ALTdm_PREFIXES:	"prefixes"
ALTdm_RUNTIME_CLASS_WORDS:	"runtime_class_words"
ALTdm_SUFFIXES:	"suffixes"
ALTdm_TIME_ITEMS:	"time_items"
ALTdm_YES_NO_SYNONYMS:	"yes_no_synonyms"

**The default, when this REGID is not used, is "list\_items". (This retains backwards compatibility.)**

**The following sample VOS/ADL code loads and uses a constrained SpeechWorks module:**

```
const REGID_SwVrLoadVocab = 2817;
const REGID_SwVrGrammarDialog = 2828;
const REGID_SwVrGrammarVocabType = 2829;
VrSetString(REGID_SwVrGrammarDialog, "Constrained_ALTdmAlphanum");
VrSetString(REGID_SwVrLoadVocab, "ITEMLIST;advoc.voc");
VrSetString(REGID_SwVrGrammarVocabType, "entry_list");
VrPlayAndRecogGrammar("Say10DigitAccountNumber.vox", "ITEMLIST", "", 0,
0, "vox,8000,8,mulaw");
```

**The Standard.cfg file would contain an entry as such:**

```
Module Constrained_ALTdmAlphanum ALTdmAlphanum
{
    restrict_to_entries: TRUE
}
```

**Problem:** In VOS 6.x, RLLs could receive driver-level events through the deventf callback function. This is broken in later versions. The callback is never called.

**Solution:** The deventf callback function was only reporting events from the legacy VOS layer. It was not reporting events that came through the new Topaz subsystem. A software correction has been made to send these events through the callback function as well. RLLs should now begin receiving deventf callback notifications of driver events again.

**Problem:** After opening an SS7 Trunk, it goes to the NetworkDown state and stays there forever. The GCEV\_UNBLOCKED event is never received from the drivers.

**Solution:** There is a caveat in the way that SS7 Trunks must now be opened to get them to work properly. The string ":L\_SS7 " must always be prepended to the device name passed to gc\_OpenEx. After making this software correction, SS7 Trunks now receive GCEV\_UNBLOCKED events from the drivers, and the Trunk Resource now transitions properly to the Idle state so it can be used.

**Problem:** The maximum precision for tone definitions is limited to tenths of a second. The hardware supports hundredths of a second, and this extra precision is required.

**Solution:** A software correction has been made to allow for the support of hundredths of a second in tone definitions. A new Profile ID has been created that can be used to indicate that all the times stated in a tone definition are in hundredths of a second rather than tenths.

Set the "ToneTimesAreHundredths" Profile ID to true to enable this functionality. Set it to false or leave it undefined to continue using tenths of a second.

For example:

```
\Tones[0]\ToneTimesAreHundredths=True
```

**Problem:** TrunkMakeCall is called with REGID\_R4GcMonitorMakeCallEvents set so that the Trunk goes immediately to the Connected state. After a while the call is disconnected by calling TrunkDisconnect. At the same time, the drivers give us a GCEV\_CONNECTED event and the following error is given: "@e Topaz error:R4GcTrunkX ResetOnBadState(), State=DEVSTATE\_Disconnecting". After this, the Trunk hangs in a bad state.

**Solution:** A software correction has been made to address the race condition of receiving a GCEV\_CONNECTED event while disconnecting the call. This event is now ignored when the Trunk is in the Disconnecting state. The return code of TrunkDisconnect can be checked, as normal, to ensure that it has completed successfully.

**Problem:** While using DCB conferencing with ADL (VOS) legacy application functions, an error similar to the following can appear in the ADL (VOS) log and the application can stop responding to conference member DTMF input:

Netslot 8 not found

This problem occurs after calling DCB\_cdel. All parties in active conferences that use the conference ID that was passed to DCB\_cdel stop receiving DTMF notification.

**Solution:** The problem was resolved with a change to the ADL DCB legacy code.

**Problem:** There was a need in CT ADE to set three SS7 MakeCallBlock indicators: satellite\_indicator, echo\_device\_indicator and continuity\_check\_indicator.

**Solution:** Three new RegIDs have been added to allow access to these MakeCallBlock indicators. The indicators may be set using the TrunkSetInt() function in Graphical VOS or the SetInt() function in VoiceBocx. These are set after opening the CT ADE Trunk Resource and before placing a call. The RegIDs are:

```
R4GcMKB7SatelliteInd    3685
R4GcMKB7EchoDeviceInd  3686
R4GcMKB7ContCheckInd   3687
```

For example, to set R4GcMKB7SatelliteInd in VOS:

```
# set no satellites
TrunkSetInt(3685, 0);
```

**Problem:** The indirect jump (jump to variable) does not behave correctly. It jumps to the wrong label and it overflows the stack, causing VOS to exit after 15 calls.

**Solution:** ADL was modified to support the correct behavior for indirect jump. It now jumps to the label stored in the referenced variable and does not overflow the stack.

**Problem:** In Graphical VOS TrunkWaitCall() returns 1 (True) when the Trunk Resource transitions to the NetworkDown state. It should return 0 (False). This is also possible for other Trunk functions but less likely.

**Solution:** TrunkWaitCall() now returns 0 (False) when a transition to NetworkDown state occurs during the operation. This is also true of other Trunk functions.

**Problem:** Web RII ftp losing connections.

**Solution:** CT ADE uses the wininet.dll functions for the Web RLL, and that the only FTP server software this is truly compatible with is MS IIS. If you're having trouble with your FTP server dropping connections (and leaving them hanging open on the server side), try updating your CT ADE system to use a more recent version of MSIE. Also, make sure you're using as recent a version of IIS as your OS will allow.

**Problem:** When CT ADE receives the Out of Service Message on R4GcTrunk, it ignores the message and does not transition to NetworkDown.

**Solution:** R4GcTrunk was modified to now transition to NetworkDown when Out of Service is received. Also, it returns to Idle when the In Service message is received.

**Problem:** The audio playback may stutter for D/41D and D/41E boards in ADL. This problem only occurs in ADL when using a parallel port dongle. Systems with USB dongle will not experience this problem.

**Solution:** In ADL audio playback on D/41D and D/41E boards experienced audio playback glitches because of starvation of the thread sending audio data to the board. This interference has been reduced or eliminated by a code change to the ADLRuntime.

**Problem:** A ResetOnBadState message may be received in CT ADE when using R4GcTrunk, e.g.:

```
Topaz error:R4GcTrunk0 ResetOnBadState(), State=DEVSTATE_NetworkDown
(42027 DEV.CPP:1184)
```

This may occur in either ADL or ADX.

**Solution:** CT ADE has been modified so that when a line device goes down with R4GcTrunk in the middle of a call, the trunk state will remain NetworkDown when the GCEV\_Disconnected event arrives after the GCEV\_Blocked event. This fix applies to both ADX and ADL.

**Problem:** libldgadminconsumerfw.dll not found error

**Solution:** This is expected in the latest version of CT ADE. The messages appear in the log but will not trigger any @E Topaz errors. It was an interim fix for HotSwap code that will be removed in the near future. They can be safely ignored. All it means is their system does not support HotSwap. (It's likely not a cPCI chassis and/or does not have SR6 installed.)

**Problem:** After passing a prompt file to MatchBocx for use as a prompt during recognition, the file is forever locked for editing. It cannot be modified or deleted.

**Solution:** A software correction has been made to fix a problem where MatchBocx was not properly closing prompt files when done with them. Because the file was considered to be open by the operating system, it could not be deleted or modified by another process. The file is now properly closed when the PlayAndRecog method is completed.

**Problem:** When starting an ISDN trace with TrunkSetString(REGID\_R4GcTraceFileName, "TraceDialogicAPI.log") in VOS/ADL or SetTrunkString(REGID\_R4GcTraceFileName, "TraceDialogicAPI.log") in CallSuite/ADX, the following error may occur:

```
@X gc_StartTrace(2[dtiB1T1], 0x0142617B)=-1,
    *tracefilename="TraceDialogicAPI.log", lib 2 gen_error 0x89
    loc_error 0x307, Bad timeslot
```

**Solution:** The problem was fixed.

**Problem:** R4GcSetParm RegID was implemented for integer parameters only, therefore an attempt to set the string parameter with this REGID returned error:

```
TrunkSetInt(508, 16385, 123)
@X gc_SetParm(4[dtiB2T1], 16385, 0x12f860)=-1, lib 2 gen_error 0x89
    loc_error 0x103, Bad argument ID
```

**Solution:** The problem was fixed by adding string parameter capability to this RegID.

**Problem:** When sending faxes with GDK (formerly the GammaFax API), users might want to change the default timeout, the number of seconds after dialing to wait for answer-tone carrier detect. (By default this timeout value is set to 30.) CT ADE did not provide any way to change this value.

**Solution:** The new integer RegID R4GrtFaxCdTimeout (2008) was added. This RegID provides direct access to the cd\_timeout field of the GFQRECORD structure.

*ADL Example:*

```
FxSetInt(2008, 45);
```

This changes the default timeout to 45 seconds.

For more on the cd\_timeout field or the GFQRECORD structure, see the GDK Version 5.0 Programming Reference Manual.

**Problem:** This error is reported in the ADL/ADX log file on startup:

```
"AD Resource Manager error:R4GcTrunk0 Internal error: invalid auto-open data type 0"
```

or

```
"AD Resource Manager error:R4GcTrunk0 Internal error: invalid auto-open data type 'Unknown'."
```

This can happen when using BRI or SPVR technology with AutoOpen Profile IDs.

**Solution:** The bug was fixed

**Problem:** When using asynchronous file I/O the fil\_open/fil\_putline functions always overwrite the old file in CT ADE 8.3.

**Solution:** This problem has been corrected.

**Problem:** ADL (VOS) crashes when calling ADO RLL function ABCIs() for a connection that does not open successfully with ABOPrv or ABOPub while in Async mode. An error similar to the following appears in the ADL (VOS) log file.

```
@E CADOConn 'Open' : tes : COM Error 0x80004005(Unspecified error ) :  
_Connection::Open()
```

Also, when calling ABOPrv() or ABOPub() if the connection attempt fails, the above error is logged but the application has no way to tell if the operation succeeded or failed. ABGetProp() should return 2 (connecting) for the state while the connection attempt is made but returns 0 (closed).

**Solution:** ADO RLL was modified to have ABGetProp() return the correct state (2 for connecting) when ABOPrv() is connecting. A correction was made to eliminate the crash on calls to ABCIs(). ABCIs() will return 0 (success) when called on a closed connection.

**Problem:** ADL logs the error "Media:n is already executing a function" and some tasks freeze. This problem is manifested by some ADL tasks hanging for no apparent reason and an error message similar to the following appearing in the ADL.log file:

```
@F
MediaPlayFile(.\Prompts\Mandarin\mprbt0001.wav,+,0,0,wav,11025,8,linear
): Media:92 is already executing a function.
```

The problem is caused when a call is made to the UnUse method for a resource by a task that does not own the Resource. So, for example, if task 0 owns Media Resource 0 and is performing a MediaPlayFile and task 1 calls MediaUnUse(0), the problem will occur. The association between Media 0 and Task 0 has incorrectly been broken by the MediaUnUse(0) call from the non-owning task 1. Task 0 hangs and the error is logged when ADL, thinking Media 0 is available, allows another task to attempt to use it even though it is busy and owned by task 0.

**Solution:** ADL was modified so that when MediaUnUse and other UnUse commands are called from a task that does not own the Resource being freed, the association between the owner task and the Resource will not be broken. An error will now be returned if you call the UnUse commands on a Resource from a task that does not own the Resource.

**Problem:** The ADO RLL function ABCIs returns success (returns a 0) but logs an error when ABCIs is called on a closed connection. The error would look like the following.

```
@E CADOConn '~CADOConn' : ABConn : COM Error
0x800a0e78(adErrObjectClosed) : Close
```

**Solution:** ADO RLL was modified to return success (return a 0) when close is called for a closed connection and to not log an error.

**Problem:** XxxGetOwnerTask() functions return -1 instead of -2 for unassigned Resources.

**Solution:** The XxxGetOwnerTask functions now return -2 when the Resource is valid but unassigned.

**Problem:** After destroying a conference without removing an MSI party first, the next time the MSI party makes a call, an error similar to the following can appear in the ADL (VOS) log:

```
212 Station is Assigned to Conference.
```

**Solution:** Each party needs to unlisten before a conference is destroyed. (The ConfRemoveParty function does this unlisten for you, but it was possible to destroy a conference without calling ConfRemoveParty.)

The ADL/VOS code has been updated so the ConfDestroy function now performs the unlisten for each party that hasn't been previously covered with ConfRemoveParty.

**Problem:** When checking for R4GcTrunk Resources, the Topaz Profile scanner crashes.

**Solution:** A software correction has been made to work around a driver fault. Certain driver functions will return before processing is complete without giving any indication that the work has not been done. This is the case when opening DM3 devices with the Cheetah drivers. The device is not fully open, but the function returns. When we

query the timeslot, and then close the device, there is a chance that the device is still not fully open. The drivers do not catch this race condition and crash due to an open and close being performed at the same time on the same device. To work around this we open all devices before checking for timeslots, and then close them all. This gives the drivers more time to finish the opening sequences.

**Problem:** When one ADL task is using a Trunk resource and another task is used to try and force the first task to disconnect the trunk, the user sees "@F TrunkAbort(<trunk number>): AD Resource Manager error:R4GcTrunk0 COMMAND\_Abort() called from invalid state DEVSTATE\_Idle (42039 DEV.CPP:1523)". Using task\_hangup() also fails: "@F task\_hangup(<task number>): 0 is not a valid Undefined resource index"

**Solution:** Unfortunately, TrunkAbort() is not meant to be used to abort a TrunkWaitCall() or TrunkWaitDisconnect() being used on a different task (From the ADL 8.3. documentation: "TrunkAbort can be used only when the Trunk Resource is in the Calling state. If you want to "abort" a TrunkWaitCall or TrunkWaitDisconnect function, use task\_hangup."). With task\_hangup, a problem was found in the way resources are tracked inside of ADL tasks, such that if the resource you are trying to abort (i.e. Trunk) is more than 0, you would see the above behavior. Once the problem was corrected, task\_hangup now behaves correctly.

**Problem:** With tracing enabled, a user attempts to use an ADL function that includes a string argument containing percent symbols ('%') embedded within it. Once that statement is executed, the ADL runtime crashes immediately.

**Solution:** An error was found inside of the ADL trace internals when updating the product to support Unicode. In that case, the '%' symbols in the string argument were being processed internally like a printf format string, causing a null pointer condition. Now, string arguments are processed correctly, such that any embedded percent symbols will not cause a crash.

**Problem:** By default both Legacy and Topaz flavors of VOS/ADL automatically call gc\_ReleaseCall after a call has been terminated to cleanup the trunk resource and prepare for the next call. This action removes any additional information about the call, such as billing information, if it is available. In Legacy, the client was able to disable automatic call release and get the information from the call before manually releasing it. There was no way to do the same in Topaz based ADL.

**Solution:** New Boolean REGID R4GcStoreEndOfCallInfo (3696) was added to Topaz to be able to collect the information and then release the call automatically if it is set to True. One caveat here is that you need to set R4GcSigInfoBufCount to 0 and use R4GcCallInfo(514) to fetch the info instead of R4GcGetSigInfo(556) which we normally recommend. This is because of the bug in gc\_GetSigInfo which doesn't always work at the end of call scenarios.

**Problem:** The fil\_write ADL function call may write extra spaces when text is written to file. This happens in a random fashion and is not related to the file system used.

**Solution:** The ADL was modified to write the text with the correct position.

**Problem:** In most cases there is just one event carrying SigInfo for the call and the application can collect the signaling information using R4GcGetSigInfo REGID. However, if there

are several events reported in the short amount of time, the application may not have a chance to get the info for all the events because the next event may overwrite the info in the current one.

Solution: New REGIDs added to be able to save/retrieve number of SigInfo events.

R4GcQueueSigInfoEnable(3691) is a boolean dynamic REGID which should be set to True if the application expects to receive several GC events carrying signaling information within a short amount of time and needs to save and process all of these events. It is set to False by default since in most cases the protocols send just one event like this per call and it can be retrieved using regular R4GcGetSigInfo

R4GcQueuedSigInfoBlkCount(3692) is an integer REGID which returns the number of SigInfo data blocks saved. One data block is saved per each qualifying event (like GCEV\_NOTIFY).

R4GcDeleteLastQueuedSigInfoBlk(3693) is an integer REGID which is used to delete the last (oldest) saved block.

R4GcRetrieveLastQueuedSigInfoBlk(3694) is a binary REGID which returns 1023 bytes of data saved for each block. The actual info can be shorter, but the application must allocate at least 1023 bytes to receive the data (use buffers in case of ADL).

R4GcQueuedSigInfoBlkOverflow(3695) is an integer REGID which reports the number of "overflows", where the maximum number of SigInfo blocks was reached and the oldest block was discarded. The application must set the appropriate maximum number of blocks to be saved using R4GcMaxParmBlks REGID (note that the same REGID is used for both signaling info blocks and IPLINK parameter blocks) and/or delete the blocks after processing them to leave room for the new events/blocks to be saved. The default number of blocks is 1. Maximum number is 256.

Example code (this example assumes that the application needs to store and process GCEV\_NOTIFY messages).

R4GcSigInfoBufCount must be set to 10 or more in the profile (it is done for the benefit of the Dialogic driver so that it can queue its own messages, it should not be confused with the R4GcQueuedSigInfoBlkCount).

```
TrunkSetBool(R4GcStoreNotifyInfo, 1);
TrunkSetBool(R4GcQueueSigInfo, 1);
...Call is established and connected...
TimesInLoop = 5; # 5seconds to wait since we sleep 1 second for each
iteration
buf = buf_use();
    # Wait for availability of signaling info blocks and get them as
they arrive
    while(TimesInLoop--)
        count = TrunkGetInt(R4GcQueuedSigInfoBlkCount ,
GCEV_NOTIFY); #GCEV_NOTIFY is 0x835 (gclib.h)
        if (count > 0)
            #get data
```

```

                                code =
TrunkGetBin(R4GcRetrieveLastQueuedSigInfoBlk, GCEV_NOTIFY, buf, 1023);
#1023 is size

                                ...buf processing here...
                                code =
TrunkSetInt(R4GcDeleteLastQueuedSigInfoBlk, GCEV_NOTIFY); #delete
curr. block

                                ..exit loop if enough data...
                                continue;

                                endif
                                sleep(10);
                                endwhile

```

**Problem:** When using VoiceBocx/ADX and using REGID\_R4DxMediaRecord2 (295) in order to record two timeslots, recording works correctly. Then when REGID\_R4DxMediaRecord2 is disabled (i.e. set to -1), any attempt to record just one timeslot fails.

**Solution:** When using SetMediaIntEx in VoiceBocx/ADX, there was a problem with setting REGID's to values less than zero. This has been fixed, and REGID\_R4DxMediaRecord2 can now be correctly set to -1 in order to properly disable the recording of two timeslots.

**Problem:** When calling MediaGetState from an ADL task that does not own the media, a "no current resource assigned" error is returned.

**Solution:** The ADL/VOS code has been corrected, and the MediaGetState function now returns the requested information without error.

**Problem:** The CT ADE help describes REGID\_ R4GcMonitorMakeCallEvents as "Set to true to go to the Connected state on the first event after initiating an outbound call." The recently added GCEV\_Dialing event, however, does not cause a transition to the Connected state.

**Solution:** The documentation here was not accurate. The Trunk Resource only goes to the Connected state after the first monitored event occurs (not after any event), so any of the following events will trigger the state change:

```

GCEV_ALERTING
GCEV_CALLSTATUS
GCEV_CONNECTED
GCEV_DISCONNECTED
GCEV_PROCEEDING
GCEV_PROGRESSING
GCEV_TASKFAIL

```

The GCEV\_DIALING event has been added to the list of monitored events, and will now also trigger the state change.

**Problem:** With a Springware board it is possible to assign a custom tone for use as a stop tone. Attempting to do this with a DM board causes an error and/or custom digits don't get detected.

**Solution:** This is a limitation of the DM3 technology. See the section on dx\_addtone in the document Compatibility Guide for the R4 API on DM3 "4.3. Voice API Function and Parameter Restrictions" for more information. It is possible to work around this limitation by using an RLL to monitor Dialogic events and dx\_Stopch to stop the media operation.

dx\_addtone() arguments "digit" and "digtype" are not supported for R4 on DM3. Therefore, mapping custom-defined GTD tones to digits (DG\_USER1 to DG\_USERn) is not supported

**Problem:** When running dmpvx.exe (a tool that's included with ADL) under CT ADE 8.3, the program throws an exception upon program termination. Depending on the size of the .VX file, the output file may or may not be truncated.

**Solution:** A bug was discovered and fixed in the dmpvx code. This is no longer an issue.

**Problem:** A REGID called R4GcTerminateOnFacility was added to CTADE 8.2 in January 2003. Not only was this REGID not migrated to CTADE 8.3, but the numeric value was reused for a different purpose, causing a compatibility problem for users of this feature.

**Solution:** Since the numeric value had been reused we issued a new numeric value for R4GcTerminateOnFacility (3697). Customers using the old value in CTADE 8.2 will need to change their application when they upgrade to CTADE 8.4.

**Problem:** The legacy PR.PAR file (used to configure custom IPF files in legacy mode) failed to work properly in ADL in CT ADE 8.3.

**Solution:** A bug was introduced in the PR.PAR reading code with the Unicode additions in CT ADE 8.3. This has been fixed and is no longer an issue.

**Problem:** Calling WebAbort() causes WebRll to crash ADL, generating an EXCEPT.RPT.

**Solution:** We found that if the timeout value was set to zero using WebSetTimeout() and subsequently WebAbort() was called that this would call InternetCloseHandle in the WinInet.dll, causing an exception to be thrown. We have corrected WebRll so that this does not occur.

**Problem:** During the Unicode changes for CTADE 8.3 an option was added to allow logging to be generated as ASCII for backwards compatibility reasons. The \*.err files (generated by ADLCompiler if there is a compilation error) and except.rpt (generated by ADLRuntime if there is a fatal exception) were always being created as Unicode.

**Solution:** These files now respect the [Log] Unicode=0 setting.

**Problem:** Some clients reported that ADL 8.3 functions (like glb\_XXX) can be slower than their 8.2 counterparts. It is usually not obvious because the delay is about 0.7 ms but it can add up if the application is calling the functions thousands of times in a loop.

**Solution:** The problem was fixed and performance levels restored.

**Problem:** Grammar that works in 8.2 produces extraneous leading zeros in 8.3.

**Solution:** The code was modified to properly honor the Prompts section DisableLeadingZeros setting in the CTADE.INI file.

**Problem:** ADO RLL not returning when no database is found in Asynchronous mode.

**Solution:** The ADO RLL was re-written to use background threads instead of using the ADO "Asynchronous" mode since it was found to sometimes actually be synchronous.

**Problem:** In order to overlap send of digits, REGID\_R4GcMonitorMakeCallEvents (529) is used to indicate that any event should cause TrunkMakeCall and ADXVoice.MakeCall to return immediately. However, when GCEV\_SETUP\_ACK is received, the function does not return.

**Solution:** A software correction has been made to cause TrunkMakeCall and ADXVoice.MakeCall to return immediately when GCEV\_SETUP\_ACK is received while MakeCall events are being monitored. This particular event had been overlooked.

**Problem:** The problem was reported on HMP1.1 system with multiple threads starting ADX instances at the same time. One or more trunks opened by ADX instances would sometimes transition from Opening to NetworkDown state although the GCEV\_UNBLOCKED event for the corresponding trunk was reported and logged. It is theoretically possible for this problem to occur on any Dialogic release and with any protocol but we've only seen it with HMP1.1.

**Solution:** There was a subtle race condition in the R4GcTrunk class which handles Global Call devices. The problem was fixed by changing the design of the code responsible for opening the devices.

**Problem:** ADL performs more slowly in CT ADE 8.3 than in CT ADE 8.2. This slower performance results in fewer tasks being concurrently executable.

**Solution:** ADL in CT ADE 8.4 was modified to eliminate problems with inefficient array and string processing.

**Problem:** The Append option for the ADL Settings file does not work. The log file is overwritten even when "Append to existing log" is selected in ADL Studio from the Logging Dialog or Append is set to 1 in the Log section of the ADL settings file.

**Solution:** The ADL was modified to now append new log file entries to the existing log file rather than overwriting it.

**Problem:** The Application calls TrunkBlock and Global Call trunk is in the Blocked state. The trunk receives GCEV\_BLOCKED event and the state changes to NetworkDown. The trunk receives GCEV\_UNBLOCKED but the trunk state remains NetworkDown.

**Solution:** CTADE was updated to include NetworkDown->Blocked state transition. The trunk will transition from NetworkDown to Blocked in the situation described above.

**Problem:** The time and date of an exception is not available in the generated except.rpt file. The timestamp of the file itself cannot be used because multiple exceptions may be appended within the same file and a date and time are needed for each one.

**Solution:** A software correction has been made to add a date and time stamp to the text generated within the except.rpt file. This now provides the required information.

**Problem:** ADO Freeze: calls ABTabNm in a loop VOS hangs and ABTabNm doesn't return.

**Solution:** This showed up intermittently due to the way Windows handles threads and orphaned critical sections. This has been fixed.

**Problem:** FaxUseNext opens 2 Fax resources.

**Solution:** CT ADE was treating resource opens as if they were all the result of auto-opens. This caused the Use built-ins to erroneously run again. This is now fixed for all Use built-ins.

**Problem:** ActiveX which works in ADL/VOS 8.2 crashes 8.3 when a string is returned by the ActiveX.

**Solution:** A software correction was made to resolve this issue.

**Problem:** Only two SS7 trunks are scanned and CT ADE resource profile regardless of the number of trunks in the system.

**Solution:** The CT ADE resource scanner was modified to now scan all SS7 trunks in a system.

**Problem:** When using SAPI4 on an HMP system we were getting events on the media device that was opened in the Sapi5VR Dev from termination events of the Media Dev's commands.

**Solution:** This problem has been resolved.

**Problem:** For the NuVR (Nuance Voice Recognition) technology, when calling the VRSetString() method in ADL or the SetString() or SetStringEx() methods in ADX with a string longer than 256 characters, the following error is generated in the log.

**Solution:** CT ADE was modified so that a maximum string length of 4096 is supported in the Nuance VR technology. Strings passed to set string methods may now be up to that length.

**Problem:** The ADORLL performs about ten times more slowly in CT ADE 8.3 vs. CT ADE 8.2. When calling any ADORLL function, that function takes approximately ten times as long to execute compared to CT ADE 8.2. This results in being able to run only a small number of tasks, ten versus one hundred for example.

**Solution:** The code in ADORLL was modified to eliminate the performance problem.

**Problem:** After a Network Hub client crashes the server must be restarted in order for clients to be able to connect or send messages.

**Solution:** Network Hub was modified to allow the Network Hub Server to close the connection with a client after the client crashes. This allows the server to continue to function normally, accepting new connections and processing client messages.

**Problem:** In CT ADE 8.3 the scanner fails to find Nuance while using Java 1.4.1. This was previously fixed in CT ADE 8.2.2.

**Solution:** CT ADE 8.4 and newer now support Java 1.4.1 as well.

**Problem:** It was not possible to set PROCEEDING as a listened to event using REGID\_R4GcSetEvtMsk(543).

**Solution:** This has been resolved.

**Problem:** Scanning of D/41JCT board results in incorrect profile when R4FxFax technology is excluded from scan. For example:

```
\Techs[R4AgTrunk]\Devs[7]\RestrictListenFaxTechType=R4FxFax  
\Techs[R4AgTrunk]\Devs[7]\RestrictListenFaxTechIndex=-1
```

**Solution:** The problem was fixed by not generating the entries for the restricted listen to fax in AgTrunk when fax is excluded from scan.

**Problem:** If an ADL application runs as a service and is stopped using the ADL service control panel it will appear to be shutdown correctly, but it will report an error to the event viewer: "The VOS Runtime Service terminated unexpectedly. It has done this 1 time(s). The following corrective action will be taken in 0 milliseconds: No action."

**Solution:** Code was fixed to let the service transition to the stop state correctly.

**Problem:** If Network Hub RLL function call is made and it doesn't return immediately, the whole ADL (all tasks) is blocked until the call returns. The problem was reported with HubMsgSend but all RLL function calls which may take time to return suspend ADL (all tasks) for the duration of the call .

**Solution:** Network Hub RLL was redesigned to execute function calls on the separate threads so that main ADL thread is not blocked.

Problem: Disconnect (hangup) tone does not interrupt TTSSpeakFile method.

Solution: The code was fixed by making the TTS device forwarding the hangup tone event notification to the appropriate media device.

Problem: Error "@E NetworkHub RLL:MsgGet(vTimeOut, &IResult)error 20016 DoMsgGet(two) cannot be invoked when Message Events are enabled" can be reported after calling HubMsgGet function with timeout.

Solution: The timeout and waiting mechanism is changed to fix the problem in Network Hub.

Problem: TTS SpeakFile doesn't return until hangup if the SAPI 5 engine goes down.

Solution: This issue has been resolved.

Problem: Device busy errors using SAPI5VR on DMV1200B.

Solution: A problem with opening multiple board handles was discovered in CT ADE and resolved.

Problem: In the Web RLL documentation for 8.3 it indicates "If WebWait is waiting for the WebGetData function to complete, WebWait returns the data received from the URL (up to the maximum number of characters set by the ADL Settings file MaxStringLength parameter)". Customer indicates it is only returning 128 characters.

Solution: The 128 characters restriction is removed and customers will be able to get up to MaxStringLength of data. URLs can now be specified by user with up to 64K in total length, and will be cut off at that length if there is an attempt to pass in one that is longer.

Problem: When ADL is started as a service, if the ADL program is stopped from the ADL control panel then the service doesn't get set to the stopped state and there is an error written to the event viewer

```
The VOS Runtime Service service terminated unexpectedly. It has done this 1 time(s).
```

Solution: The service shutdown code was redesigned to fix this problem.

Problem: Customer is adding a party (from their scx adapter) to a conference via regID 1917 REGID\_R4ConfAddTimeslot. After the timeslot is added they want to change it from ListenOnly to full duplex using ConfSetInt(1904,PartyIndex,1). Customer used a reasonable party index but topaz returns an error "Topaz error:R4MsConf1 Conference party not found Party index 2 out of range (17009 ConfDev.cpp:604)".

Solution: A new regid R4ConfPartySCBAttrBits (1918) is added to let customer change the attribute of the timeslot based on time slot number (as opposed to conference party index). For example, use ConfSetInt(1918, 8, 1) to set timeslot 8's attribute to 1.

**Problem:** On Denali series boards, when the user hangs up the phone on an MStTrunk while the dialtone is being played, the hangup does not stop the dialtone playing so the subsequent calls are not detected.

**Solution:** The Media Resource will now be aborted if it is playing a tone and the MSI Trunk is aborted.

Two new REGIDs were also created:

To keep the battery from being turned off at all during a TrunkDisconnect, please set REGID\_R4MsDisconnectByBatteryOff (328) to false via the TrunkSetBool function.

If you wish to keep the battery-off logic (if it is required to hang up), but want it to work correctly, you will need to increase the amount of time the battery is off prior to it being turned back on. It looks like for your system battery is not off long enough prior to it being turned on and this is causing a failure. You should set REGID\_R4MsDisconnectMs (329) to a value that is probably around 2000. The default of 1000 does not seem to be long enough on your system.

**Problem:** Setting code page locale in ADL

**Solution:** New section [Localization] was added to ADL.INI file. The new key for this section is called "locale". In order to set the new locale the section may look like this:

```
[Localization]
locale=English
```

This results in ADL calling `_tsetlocale( LC_ALL, "English" );` on startup. See MSDN help Run-Time library Reference for more information about `_tsetlocale` function.

**Problem:** Sometimes the information written to the log file isn't in the correct order.

**Solution:** A fix has been made to guarantee `voslog()` to be atomic. Individual strings passed as different arguments should no longer be split in the log.

**Problem:** Memory leak in Web RLL

**Solution:** The memory leak and one race condition have been fixed in Web RLL.

**Problem:** When using an RLL to handle call diversion information using the DPNSS protocol, virtual calls in the form of offered events trigger `ResetOnBadState`.

**Solution:** A new REGID called REGID\_R4GcIgnoreVirtualCalls (4228) has been added to allow CT ADE to ignore these virtual calls. It is the responsibility of the RLL to handle the call diversion information.

**Problem:** Only half of Chinese character is displayed in ADL/ADL Studio.

**Solution:** Some issues involving fonts in Asian languages were detected and resolved.

**Problem:** gettaskvar returns incorrect result in 8.3.

**Solution:** gettaskvar was returning the number of character in the variable instead of zero to indicate success. This has been corrected.

**Problem:** Fax polling using the examples in CTADE help manual may result in error message "The previous command 'COMMAND\_NONE' has not reached its completion state DEVSTATE\_Idle" and/or fax transaction being aborted.

**Solution:** The problem was fixed. New REGID R4FxFaxTermMsk = 1714 added to be able to query the fax termination reason. Here is the example:

**Outbound:**

```
#call established
    FaxReceive("c:\faxrec.tif",2); #start poll to receive file
```

**Inbound:**

```
#call established
    #This means "receive with poll allowed, if term. reason is poll
received, then send fax".
    FaxReceive("fpoll.tif", 1);
    code = FaxGetInt(1714); #R4FxFaxTermMsk
    switch (code)
    case 1:    #TM_FXTERM, see faxlib.h
        voslog("normal fax received");
    case 2:    #TM_POLLED, see faxlib.h
        voslog("poll request, send them a fax");
        FaxAddTiffFile("small.tif");
        FaxSend();
    case 4:    #TM_VOICEREQ, see faxlib.h
        voslog("voice");
    default:
        voslog("unknown termination reason= ", code);
        exit(1);
    endswitch
```

**Problem:** WebRLL crash

**Solution:** The order of the *InternetCloseHandle* function calls was incorrect in the RLL. In order to use wininet calls, you are supposed to open the general connection handle first with *InternetOpen*. Then you use *InterentConnect* to start the FTP or HTTP session which gives you back another handle. Naturally, the order of closing the handles must be reversed: first close the handle from the *InterentConnect* and then the handle from the *InternetOpen*. WebRLL was closing the main handle first and then the call to close the connection handle was failing if wininet had time to fully close the main handle.

A handle leak in the PutFile thread was also fixed.

Problem: ActiveX events were causing "@E 668" in 8.3.

Solution: This has been corrected.

Problem: itoc and ctoi function always use Unicode.

Solution: An optional argument was added to each function to allow extended ASCII to be used instead for compatibility reasons.

Problem: A bug was introduced which caused "TTS\_E\_NETWORK\_PROBLEM" when using RealSpeak 3.5 in standard mode if the EnginePath was set in the CTADE.INI file.

Solution: This problem has been corrected.

Problem: Getting ANI failing when receiving overlap DNIS.

Solution: A new REGID called REGID\_R4GcGetANIONMoreDigits (4229) has been added to R4GcTrunk. If they set this bool to true, the gc\_GetANI call will only be done after the GCEV\_MOREDIGITS event is received.

Problem: The application crashes with MediaPlayerString & MediaPlayerMoney.

Solution: Two memory leaks were found and fixed.

Problem: Network Hub/rll - "Internal Error. Instance table is full" and memory leaks.

Solution: Extensive changes were made to the threading model in this RLL.

Problem: RegID 541 - gc\_getlinedevstate error on DM board

Solution: The problem here is that when you open a very first channel on some board, you implicitly open the board itself. Then, if you try to close and re-open the board level device, it gives an error, saying "Cannot open the same resource more than once in the same process". It's a limitation of Intel Dialogic.

CT ADE has been modified to only open the board device only once, and then re-use the handle.

Problem: Active X crashes in 8.3 works in 8.2

Solution: Bugs found in variant to string conversion have been fixed.

Problem: NetHub RLL unnecessarily logging

Solution: Network Hub was logging detailed trace data when logging was set to normal or off. This has been corrected.

Problem: Version 8.3 and newer of CTADE ship with include files. These files are meant for inclusion in your ADX or ADL project. They define language specific constants for

REGIDs. CTADE users are recommended to use these constants instead of the numerical values in order to create more readable, self-documenting code. It was discovered that none of the files were usable with C#.

**Solution:** A new include file named REGIDS.CS has been created. This file will be included in all new patches and installers from the date of introduction and will be automatically generated as the other language include files are.

**Problem:** When using a SAPI4 TTS resource with CT ADE 8.3 the application will get the following error when setting the TTS personality:

```
@b TtsSetPersonality(SAPI4_Person0)
```

```
040921 100805.73 main:0066[1] @F TtsSetPersonality(SAPI4_Person0): AD  
Resource Manager error:SapiTTS0 Could not select text-to-speech mode (159013  
SapiTTS.cpp:524)
```

**Solution:** The problem is that the default Audio Format settings for the SAPI 4 are incorrect. The default settings in the profile are:

```
\Techs[SapiTTS]\Devs[x]\AudioFormatCoding=Mulaw
```

but CT ADE only supports: 8-bit 8KHz Linear PCM Audio Format for SAPI4 TTS

So you will need to change all of the SapiTTS entries from:

```
\Techs[SapiTTS]\Devs[x]\AudioFormatCoding=Mulaw
```

to

```
\Techs[SapiTTS]\Devs[x]\AudioFormatCoding=Linear
```

## Notes, Known Bugs, and Workarounds

---

### CT ADE Installation

**Issue:** When a user modifies a Minimal Installation the changes are not recorded by the installer. If later the user tries to repair the installation the Minimal Installation will be restored and if the user removes the installation the components that were added to the minimal installation are not removed.

**Solution:** If you select the Minimal installation, you should not manually modify the installation components. To select the installation you want, choose the Custom installation.

---

### Resource Manager

#### Supervised and Blind Transfers: New Profile Entries

**Issue:** Two new Profile IDs have been added

```
TransferBlindSupported  
TransferSupervisedSupported
```

If you're upgrading from CT ADE 8.2, please note that these Profile entries must be set to True for your existing blind and supervised transfer code to continue working.

**Solution:** The Profile scanner will automatically set these Profile IDs for you when you recreate the Profile. See "Recreate Profile" in the ADL or ADX user's guide for more information on recreating the Profile.

### **RoutingRestrictionsDisabled**

**Issue:** The ADX and ADL help do not describe the new Profile ID RoutingRestrictionsDisabled.

**Solution:** A description of the Profile ID follows.

The RoutingRestrictionsDisabled Profile ID lets you override any routing rules that are set up in the Profile. For example, if your Profile contained the following entries

```
\Techs[R4AgTrunk]\Devs[3]\RoutingRestrictionsDisabled=True  
\Techs[R4AgTrunk]\Devs[3]\ListenType=Fixed  
\Techs[R4AgTrunk]\Devs[3]\FixedListenTechType=R4DxMedia  
\Techs[R4AgTrunk]\Devs[3]\FixedListenTechIndex=0
```

the ListenType, FixedListenTechType, and FixedListenTechIndex entries would be ignored and your application would be free to route R4AgTrunk Resource 3 to any available Media Resource.

As you can see in this example, RoutingRestrictionsDisabled applies to a single Resource. The Profile ID is valid for any CT ADE Technology. By default, RoutingRestrictionsDisabled is set to False.

#### **Note**

In most cases, you won't need to override the routing restrictions that have been automatically set up in the Profile. You'll only use RoutingRestrictionsDisabled if your application is manually controlling the Resource routing.

See "Dependencies," "Fixed Listen Resources," "Free Listen Resources," and "Profile IDs Used for Routing" in the ADL or ADX user's guide for more on routing and routing rules.

### **REGID\_R4GcSetupParmIDs**

**Issue:** The ADX and ADL help do not sufficiently describe REGID\_R4GcSetupParmIDs.

**Solution:** A more complete description follows.

This RegID sets up the arguments for the Global Call API functions gc\_util\_insert\_parm\_val and gc\_util\_insert\_parm\_ref. See the *Global Call API Software Reference* for a description of these API functions, their usage, and their arguments.

REGID\_R4GcSetupParmIDs requires three values, corresponding to the gc\_util\_insert\_parm\_val and gc\_util\_insert\_parm\_ref parameters setID, parmID, and data\_size.

REGID\_R4GcSetupParmIDs sets up the parameters used by the gc\_util\_xxx functions, but it does not call those functions. After setting the parameters with REGID\_R4GcSetupParmIDs, you need to use one of the following RegIDs to pass the data to the gc\_util\_insert\_parm\_val or gc\_util\_insert\_parm\_ref function, which inserts the data into the GC\_PARM\_BLK:

REGID\_R4GcInsertParmVal  
REGID\_R4GcInsertParmRef  
REGID\_R4GcFindParm  
REGID\_R4GcInsertStruct.

#### *ADL Example*

In ADL you use TrunkSetInt with the following parameters:

```
TrunkSetInt(REGID_R4GcSetupParmIDs, setID, parmID, data_size)
```

So if you want to set the IPPARM\_DISPLAY, you could use the following code:

```
# Set up the GC_PARM_BLOCK.  
TrunkSetInt(REGID_R4GcSetupParmIDs, IPSET_CALLINFO, IPPARM_DISPLAY,  
            AUTO_SIZE);  
TrunkSetString(REGID_R4GcInsertParmRef, "User's display string");  
  
# Use GC_PARM_BLOCK.  
TrunkSetInt(REGID_SetUserInfo, GCTGT_GCLIB_CHAN, GC_SINGLECALL);
```

#### *ADX Example*

In ADX you use SetTrunkIntEx with the following parameters:

```
SetTrunkIntEx(REGID_R4GcSetupParmIDs, data_size, 2, setID, parmID)
```

Remember that the value 2 in the third parameter specifies that there are two more parameters following—this value is not passed through to any API function.

To set the IPPARM\_DISPLAY, you could use the following code:

```
' Set up the GC_PARM_BLOCK.
```

```

Call ADXVoice1.SetTrunkIntEx(REGID_R4GcSetupParmIDs, AUTO_SIZE, 2,
    IPSET_CALLINFO, IPPARM_DISPLAY)

Call ADXVoice1.SetTrunkStringEx(REGID_R4GcInsertParmRef, "User's
    display string")

' Use GC_PARM_BLOCK.

Call ADXVoice1.SetTrunkIntEx(REGID_SetUserInfo, GCTGT_GCLIB_CHAN, 1,
    GC_SINGLECALL)

```

## REGID\_R4GcExtension

**Issue:** The ADX and ADL help do not sufficiently describe REGID\_R4GcExtension.

**Solution:** A more complete description follows.

This RegID calls the Global Call API `gc_Extension` function. See the *GlobalCall API Software Reference* for a description of this API function, its usage, and its arguments.

Calling REGID\_R4GcExtension sets the following `gc_Extension` parameters: `target_type`, `target_id`, `ext_id`, and `mode`.

Because `gc_Extension` is often used with the GC\_PARM\_BLOCK, you'll probably need to use other RegIDs to set up the parameter block before calling REGID\_R4GcExtension.

### ADL Example

In ADL you use `TrunkSetInt` with the following parameters:

```
TrunkSetInt(REGID_R4GcExtension, target_type, target_id, ext_id, mode)
```

This example sends a non-standard message using REGID\_R4GcExtension:

```

# Set up the GC_PARM_BLOCK.
TrunkSetInt(REGID_SetupParmIDs, IPSET_MSG_H245, IPPARM_MSGTYPE,
    INTSIZE);
TrunkSetInt(REGID_InsertParmVal, IP_MSGTYPE_H245_COMMAND);
TrunkSetInt(REGID_SetupParmIDs, IPSET_NONSTANDARDATA,
    IPPARM_NONSTANDARDATA_OBJID, AUTO_SIZE);
TrunkSetString(REGID_InsertParmRef, "12345");
TrunkSetInt(REGID_SetupParmIDs, IPSET_NONSTANDARDATA,
    IPPARM_NONSTANDARDATA_DATA, AUTO_SIZE);
TrunkSetString(REGID_InsertParmRef, "NSC_Chan1");

```

```
# Call gc_Extension to send the message.
TrunkSetInt(REGID_R4GcExtension, GCTGT_GCLIB_CRN, AUTO_ID,
            IPEXTID_SENDMSG, SYNC);
```

### **ADX Example**

In ADX you use SetTrunkIntEx with the following parameters:

```
SetTrunkIntEx(REGID_R4GcExtension, mode, 3, target_type, target_id, ext_id)
```

Remember that the value 3 in the third parameter specifies that there are three more parameters following—this value is not passed through to any API function.

To send a non-standard message, you could use the following:

```
' Set up the GC_PARM_BLOCK.
Call ADXVoice1.SetTrunkIntEx(REGID_SetupParmIDs, INTSIZE, 2,
                            IPSET_MSG_H245, IPPARM_MSGTYPE)
Call ADXVoice1.SetTrunkIntEx(REGID_InsertParmVal,
                            IP_MSGTYPE_H245_COMMAND)
Call ADXVoice1.SetTrunkIntEx(REGID_SetupParmIDs, AUTO_SIZE, 2,
                            IPSET_NONSTANDARDDDATA, IPPARM_NONSTANDARDDDATA_OBJID)
Call ADXVoice1.SetTrunkStringEx(REGID_InsertParmRef, "12345")
Call ADXVoice1.SetTrunkIntEx(REGID_SetupParmIDs, AUTO_SIZE, 2,
                            IPSET_NONSTANDARDDDATA, IPPARM_NONSTANDARDDDATA_DATA)
Call ADXVoice1.SetTrunkStringEx(REGID_InsertParmRef, "NSC_Chan1")

' Call gc_Extension to send the message.
Call ADXVoice1.SetTrunkIntEx(REGID_R4GcExtension, SYNC, 3,
                            GCTGT_GCLIB_CRN, AUTO_ID, IPEXTID_SENDMSG)
```

### **REGID\_R4GcIplpCapability**

**Issue:** The ADX and ADL help do not sufficiently describe REGID\_R4GcIplpCapability.

**Solution:** A more complete description follows.

This RegID fills the Global Call IP\_CAPABILITY data structure and calls the Global Call gc\_SetUserInfo function in one step. See the *GlobalCall IP Technology User's Guide for Linux and Windows* for details on the IP\_CAPABILITY structure and the *GlobalCall API Software Reference* for a description of gc\_SetUserInfo.

You can use REGID\_R4GcIplpCapability to set up to 6 IP\_CAPABILITY parameters: capability, type, direction, payload\_type, frames\_per\_pkt, and vad.

The capability parameter must be present any time you call REGID\_R4GcIpIpCapability, but the rest of the arguments are optional. If the optional parameters are not specified, the defaults are

direction	IP_CAP_DIR_LCLTRANSMIT
payload_type	IP_USE_STANDARD_PAYLOADTYPE
type	GCCAPTYPE_AUDIO
extra.audio.frames_per_pkt	1
extra.audio.VAD	0

All the parameters can also be set individually before calling REGID\_R4GcIpIpCapability using the following RegIDs:

REGID\_R4GcIPCapAudioCap  
 REGID\_R4GcIPCapDirection  
 REGID\_R4GcIPCapPayload  
 REGID\_R4GcIPCapType  
 REGID\_R4GcIPCapFramesPerPkt  
 REGID\_R4GcIPCapVAD

#### *ADL Usage*

In ADL you use TrunkSetInt with the following parameters:

```
TrunkSetInt(REGID_R4GcIpIpCapability[, type][, direction]
            [, payload_type] [, frames_per_pkt][, vad], capability)
```

Note that the capability parameter is not optional, and it is always the last parameter in the list. The optional parameters must always be specified in the given order. If you want to use an optional parameter, you must specify all optional parameters to the left of it. For example, if you need to specify direction, you must specify type as well:

```
TrunkSetInt(REGID_R4GcIpIpCapability, type, direction, capability)
```

#### *ADL Example*

```
TrunkSetInt(REGID_R4GcIpIpCapability, GCCAPTYPE_AUDIO,
            IP_CAP_DIR_LCLTRANSMIT, GCCAP_AUDIO_g7231_5_3k);
```

Note that using REGID\_R4GcIpIpCapability like this is equivalent to the following calls:

```
TrunkSetInt(REGID_R4GcIPCapAudioCap, GCCAP_AUDIO_g7231_5_3k);
TrunkSetInt(REGID_R4GcIPCapType, GCCAPTYPE_AUDIO);
```

```
TrunkSetInt(REGID_R4GcIPCapDirection, IP_CAP_DIR_LCLTRANSMIT);
TrunkSetInt(REGID_SetupParmIDs, GCSET_CHAN_CAPABILITY,
             IPPARM_LOCAL_CAPABILITY, AUTO_SIZE);
TrunkSetInt(REGID_InsertStruct, AUTO_VALUE);
TrunkSetInt(REGID_SetUserInfo, GCTGT_GCLIB_CRN, GC_SINGLECALL);
```

### ***ADX Usage***

In ADX you use `SetTrunkIntEx` with the following parameters:

```
SetTrunkIntEx(REGID_R4GcIpIpCapability, capability, parameter_count,
              [type, direction, payload_type, frames_per_pkt])
```

Here `parameter_count` is set to the number of optional parameters that follow—this value is not passed through to any API function.

Note that ADX methods allow up to 5 optional parameters, therefore you cannot specify a VAD value using this method. If you need to change the default value for VAD, use `REGID_R4GcIPCapVAD` first to change it and then call `REGID_R4GcIpIpCapability` with the other parameters you want.

### ***ADX Example***

```
Call ADXVoice1.SetTrunkIntEx(REGID_R4GcIpIpCapability,
                             GCCAP_AUDIO_g7231_5_3k, 2, GCCAPTYPE_AUDIO,
                             IP_CAP_DIR_LCLTRANSMIT)
```

### ***ADX Example: Changing the VAD Value***

```
Call ADXVoice1.SetTrunkInt(REGID_R4GcIPCapVAD, GCPV_ENABLE)
Call ADXVoice1.SetTrunkIntEx(REGID_R4GcIpIpCapability,
                             GCCAP_AUDIO_g7231_5_3k, 2, GCCAPTYPE_AUDIO,
                             IP_CAP_DIR_LCLTRANSMIT)
```

## **REGID\_R4MsSetBoardParm**

**Issue:** The ADX and ADL help do not sufficiently describe `REGID_R4MsSetBoardParm`.

**Solution:** A more complete description follows.

This `RegID` calls the `ms_setbrdparm` API function, which sets a board-level parameter. See the *MSI/SC Software Reference* for a description of this API function, its usage, and its arguments.

**Note:** `REGID_R4MsSetMS_NCB` was deprecated and replaced with `REGID_R4MsSetBoardParm`.

The number of parameters you use when you call `REGID_R4MsSetBoardParm` depends on the board-level parameter you're setting.

Note: In the following syntax examples, the board-level parameter names are shown for readability, but in your ADL or ADX code you'll need to use integer values for these parameters.

#### *MSG\_DISTINCTRNG*

Value = 35

This API parameter initializes distinctive ringing and associates the cadence ID with it.

#### ADL syntax:

```
TrunkSetInt(REGID_R4MsSetBoardParm, MSG_DISTINCTRNG, cadence)
```

#### ADX syntax:

```
SetTrunkIntEx(REGID_R4MsSetBoardParm, cadence, 1, MSG_DISTINCTRNG)
```

ADX Note: Remember that the value 1 in the third parameter specifies that there is one parameter following—this value is not passed through to any API function.

In both ADL and ADX, you use the cadence parameter to select one of the cadence patterns that has been defined for the `ms_setbrdparm` API function.

ID	Cadence Pattern	ID	Cadence Pattern
1	MS_RNGA_TWOSEC	5	MS_RNGA_SPLASH3
2	MS_RNGA_ONESEC	6	MS_RNGA_SPLASH4
3	MS_RNGA_SPLASH1	7	MS_RNGA_LONGSHORT
4	MS_RNGA_SPLASH2	8	MS_RNGA_SHORTLONG

The cadence ID not passed directly to the `ms_setbrdparm` API function: when you set the `RegID`, the underlying code translates the cadence value to the hexadecimal value that `ms_setbrdparm` requires for the corresponding cadence pattern. See "Table 1. Ring Cadence Group A" in the *MSI/SC Software Reference* for more information on these cadence patterns.

#### *MSG\_DBOFFTM*

Value = 35

This API parameter defines the minimum length of time (in 50 ms units) before an off-hook transition is detected. See "`ms_setbrdparm`" in the *MSI/SC Software Reference* for more on this parameter.

**ADL syntax:**

```
TrunkSetInt(REGID_R4MsSetBoardParm, MSG_DBOFFTM, time)
```

**ADX syntax:**

```
SetTrunkIntEx(REGID_R4MsSetBoardParm, time, 1, MSG_DBOFFTM)
```

In both ADL and ADX, the time parameter is passed directly to the `ms_setbrdparm` API function.

***MSG\_DBONTM***

Value = 34

This API parameter defines the minimum length of time (in 50 ms units) before an on-hook transition is detected. See "ms\_setbrdparm" in the *MSI/SC Software Reference* for more on this parameter.

**ADL syntax:**

```
TrunkSetInt(REGID_R4MsSetBoardParm, MSG_DBONTM, time)
```

**ADX syntax:**

```
SetTrunkIntEx(REGID_R4MsSetBoardParm, time, 1, MSG_DBONTM)
```

In both ADL and ADX, the time parameter is passed directly to the `ms_setbrdparm` API function.

***MSG\_MAXFLASH***

Value = 37

This API parameter defines a maximum length of time (in 50 ms units) for a station to be in an on-hook state before a hook flash signal is detected. See "ms\_setbrdparm" in the *MSI/SC Software Reference* for more on `MSG_MAXFLASH`.

**ADL syntax:**

```
TrunkSetInt(REGID_R4MsSetBoardParm, MSG_MAXFLASH, time)
```

**ADX syntax:**

```
SetTrunkIntEx(REGID_R4MsSetBoardParm, time, 1, MSG_MAXFLASH)
```

In both ADL and ADX, the time parameter is passed directly to the `ms_setbrdparm` API function.

#### *MSG\_MINFLASH*

Value = 36

This API parameter defines a minimum length of time (in 50 ms units) for a station to be in an on-hook state before a hook flash signal is detected. See "ms\_setbrdparm" in the *MSI/SC Software Reference* for more on `MSG_MINFLASH`.

#### ADL syntax:

```
TrunkSetInt(REGID_R4MsSetBoardParm, MSG_MINFLASH, time)
```

#### ADX syntax:

```
SetTrunkIntEx(REGID_R4MsSetBoardParm, time, 1, MSG_MINFLASH)
```

In both ADL and ADX, the time parameter is passed directly to the `ms_setbrdparm` API function.

#### *MSG\_PDRNGCAD*

Value = 52

This API parameter selects one of the predefined ring cadence patterns on the MSI board. See "ms\_setbrdparm" in the *MSI/SC Software Reference* for more on `MSG_PDRNGCAD`.

#### ADL syntax:

```
TrunkSetInt(REGID_R4MsSetBoardParm, MSG_PDRNGCAD, cadencepattern)
```

#### ADX syntax:

```
SetTrunkIntEx(REGID_R4MsSetBoardParm, cadencepattern, 1, MSG_PDRNGCAD)
```

In both ADL and ADX, the `cadencepattern` must be an integer from 1 to 6. This value is passed directly to the `ms_setbrdparm` API function. The default value is 6, which specifies a cadence pattern of 2 seconds on followed by 4 seconds off. More values are listed in the "ms\_setbrdparm" topic in the *MSI/SC Software Reference*.

#### *MSG\_ZIPENA*

Value = 39

This API parameter sets the zip tone setting. See "ms\_setbrdparm" in the *MSI/SC Software Reference* for more on `MSG_ZIPENA`.

**ADL syntax:**

```
TrunkSetInt(REGID_R4MsSetBoardParm, MSG_ZIPENA, MS_ZIPENABLE)  
TrunkSetInt(REGID_R4MsSetBoardParm, MSG_ZIPENA, MS_ZIPDISABLE)
```

**ADX syntax:**

```
SetTrunkIntEx(REGID_R4MsSetBoardParm, MS_ZIPENABLE, 1, MSG_ZIPENA)  
SetTrunkIntEx(REGID_R4MsSetBoardParm, MS_ZIPDISABLE, 1, MSG_ZIPENA)
```

**MS\_ZIPENABLE (0)** enables zip tone generation. **MS\_ZIPDISABLE (1)** disables zip tone generation. Default = **MS\_ZIPENABLE**

**MSCB\_ND**

Value = 2

This API parameter defines the notify-on-add tone, the tone used to notify conference parties that a party has joined or left the conference.

When you set **REGID\_R4MsSetBoardParm** with the **MSCB\_ND** parameter, you also must pass in the volume, tone, duration and rfu values. These values are used to populate the **MS\_NCB** structure and then call the **ms\_setbrdparm()** function. See "ms\_setbrdparm" in the *MSI/SC Software Reference* for more on **MSCB\_ND** and the **MS\_NCB** structure.

Setting **MSCB\_ND** requires 4 additional parameters. The rfu (reserved for future use) must set to zero.

**ADL syntax:**

```
TrunkSetInt(REGID_R4MsSetBoardParm, MSCB_ND, volume, tone, duration, 0)
```

**ADX syntax:**

```
SetTrunkIntEx(REGID_R4MsSetBoardParm, 0, 4, MSCB_ND, volume, tone,  
duration)
```

**MSCB\_ZIP**

Value = 4

This API parameter defines the zip tone, the tone generated to notify a party that they are about to be connected with a call.

When you set REGID\_R4MsSetBoardParm with the MSCB\_ZIP parameter, you also must pass in the volume, tone, duration and rfu values. These values are used to populate the MS\_NCB structure and then call the ms\_setbrdparm() function. See "ms\_setbrdparm" in the *MSI/SC Software Reference* for more on MSCB\_ZIP and the MS\_NCB structure.

Setting MSCB\_ZIP requires 4 additional parameters. The rfu (reserved for future use) must set to zero.

**ADL syntax:**

```
TrunkSetInt(REGID_R4MsSetBoardParm, MSCB_ZIP, volume, tone, duration,  
0)
```

**ADX syntax:**

```
SetTrunkIntEx(REGID_R4MsSetBoardParm, 0, 4, MSCB_ZIP, volume, tone,  
duration)
```

---

## Profile Configuration Utility

**Issue:** Under some conditions, the Profile Configuration utility may not delete its temporary files.

**Solution:** In most cases, the Profile Configuration utility deletes its temporary files when it closes. There is no danger in leaving the files intact, but you can manually delete the following temporary files after saving your work and closing the Profile Configuration utility:

```
ProfileScan.txt  
ProfileTemp.txt  
ProfileBackup.txt  
ProfileProgress.txt  
ProfileDeltaBackup.txt
```

**Issue:** When opening the Profile, the Profile Configuration utility shows a blank window (the Profile window appears empty).

**Solution:** First, make sure you have started the Profile Configuration utility from the Windows\* Start menu. If you still have this problem, select "Recreate Profile..." from the File menu.

---

## ADL

**Issue:** Upgrading from VOS 8.2 without adding memory may result in slower performance.

**Solution:** With support for Unicode and strings longer than 127 characters, ADL version 8.3 requires more memory than previous versions. When you upgrade to version 8.3, you should re-evaluate your system and be sure it has enough memory to support your ADL applications. Remember to avoid relying on virtual memory instead of RAM, since swapping memory to a disk can also cause performance problems.

**Issue:** Asynchronous mode fails for Conference Resources. Attempts at using the ConfWait function will return an error:

```
ConfWait(): No asynchronous call has been made. WaitAsync() has failed.
```

**Solution:** At the time of writing, all current conferencing technologies add and remove parties instantly, so asynchronous mode is not necessary.

**Issue:** Doublewide characters (e.g. Japanese) may not display properly in the ADL Studio Editor.

**Solution:** In Windows, set the default Regional Input Locale to a locale that provides support for the characters you want to use. From the Windows Control Panel, double-click the Regional Options icon to open the Regional Options dialog. On the Input Locales tab, choose an appropriate Input Language. (In this example, you'd set the Locale to Japanese.) Set this language as your default (using the Set as Default button.) When you change this setting, you will need to reboot your PC.

**Issue:** When using doublewide characters in the ADL Studio Editor, only the first half of each line is displayed.

**Solution:** Try using a different font or in some locales, there is an option to display double-width fonts in "narrow" mode.

**Issue:** Usually after rebooting an HDSI system, you may see intermittent errors when executing TrunkUse on an HDSI trunk:

```
@E AD Resource Manager error:R4MsTrunk## Invalid state transition  
DEVSTATE_Idle->DEVSTATE_Idle
```

**Solution:** These errors can be ignored.

**Issue:** Windows Explorer\* cannot find ADL Studio. Double-clicking a .vs or .gvp file (or another file that should open in ADL Studio) in Explorer brings up the Open With dialog or the Program not Found dialog.

**Solution:** If the Open With dialog opens:  
Select ADL Studio Application from the Programs list. Check the box that says, "Always use the selected program to open this kind of file," then click OK. Explorer\* associates the file type with ADL Studio.

If the Program not Found dialog opens:

Click Locate... In the Browse for Folder dialog, browse to the directory that contains the ADL Studio application, the ADL/Bin subdirectory of the CT ADE installation directory. If you followed the default installation, you should browse to this directory:

```
C:\Program Files\Envox\CT ADE\ADL\Bin
```

Click OK to close the Browse for Folder dialog, then click OK to close the Program not Found dialog.

Issue: Memory leak and/or connection issues may appear when using database providers with the ADO RLL.

Solution: To overcome issues that appear with the OLE DB provider for Oracle\* (the version that comes with Oracle 9i v9.2\*), install version v9.2.0.2, which is downloadable from the Oracle Web site.

Testing has shown connection issues with Unicode strings and the Microsoft\* OLE DB provider for Oracle that comes with MDAC2.7. If possible, avoid using this DB provider.

Issue: The ADO RLL asynchronous mode description needs to be elaborated.

Solution: The following note should be added to the end of the "Asynchronous ADO Operations" topic in the *ADL User's Guide*:

"There is an exception to this rule on blocking operations in 8.3. Functions that do block ADL are those that begin AsMv and AsGet. These can block all ADL tasks even in asynchronous mode. If this causes a problem, try using a client cursor instead of a server cursor to speed up the operations."

Issue: The hex addresses returned by dmpvx do not match the log file.

Solution: Add 3 to the dmpvx value to get the number that corresponds to the log file.

### Flowcharter

Issue: Flowcharter cell names do not allow Unicode non-ASCII characters.

Solution: To work around this issue, limit your cell names to include only ASCII characters.

Issue: After uninstalling version 8.2, installing version 8.3, and opening the 8.2 Voicemail Flowcharter sample application, the ADL Studio cannot find the bitmaps associated with the Mailbox cells in the chart.

Solution: Although the bitmaps do not appear, the sample still runs.

---

### ADX

Issue: CallSuite 8.2 applications won't compile with the new ADX controls.

Solution: How you update your applications depends on how the control was created:

*Case 1: VB/VC++/Delphi - The CallSuite control was created by dropping on the form.*

Delete the old CallSuite control from the form, then drop the new ADX control on the form. Finally, rename the ADX control to the name of the old CallSuite control. For example, change the name of the "ADXVoice1" control to "VoiceBocx1".

*Case 2: The CallSuite control was created by referencing the .DLL directly.*

Change the path and the name from the CallSuite control to the new ADX control. For example, change "C:\Program Files\Intel CT ADE\CallSuite\Bin\VoiceBocx.dll" to "C:\Program Files\Envox\CT ADE\ADX\Bin\ADXVoice.DLL".

*Case 3: The CallSuite control was created by referencing the GUID.*  
No changes are necessary.

*Case 4: The CallSuite control was created by referencing the Class Name.*  
Change the class name from the CallSuite control name to the new ADX control name. For example, change the class name from "Intel.VoiceBocx" to "Envox.ADXVoice".

## Previous Fixes

The following problems were fixed in previous versions of CT ADE.

---

### CT ADE 8.3

#### Playing Strings

**Problem:** Leading zeros are ignored by the ADL MediaPlayerString function and the ADX Voice Control PlayString method. For example, the following function speaks only "one two":

```
MediaPlayerString("0012");
```

The expected result would be the full string: "zero zero one two"

**Solution:** The ADL MediaPlayerString function and the ADX Voice Control PlayString method now speak leading zeroes by default.

If you'd like to disable leading zeroes in strings, set the new DisableLeadingZeros entry in the Prompts section of CTADE.ini. See "Prompts Section" in the ADL or ADX User's Guide for details.

#### Additional Fixes

For details on these issues, see the corresponding section on the Envoy Technical Support web page:

Problem	Note Posted
NUANCE_LICENSE_CHECK_FAILED error when Nuance License manager runs on the remote PC	8/29/2003
Error "cannot open new thread" or other resource leak symptoms with Web RLL	8/29/2003
Global Call Trunk state changes from Opening to NetworkDown instead of Idle when hardware configuration and connection is OK	8/29/2003
Messages sent to task are missed when chain() is used	8/29/2003
Problem recording from 2 timeslots when using DM3 board	8/29/2003
Collecting DTMF Digits From An MSI Station When Connected To An MSI-Based Conference Resource	6/19/2003
REGID_R4AgD42BrdStatus=2305 Causes "Invalid Or Wrong Device Handle"	6/19/2003

Error	
Task_hangup Fails To Cause Jump To Onhangup Routine	6/18/2003
@E Topaz Error: SWVR1 Could Not Set ECCH_ADAPTMODE To 0 When Using SpeechWorks* VR	6/18/2003
"Not Enough Memory To Register Converter Factory" Error When Using VoiceBocx With Visual Basic*	6/18/2003
Error gc_GetResourceH "Invalid Parameter" On DM Boards	6/18/2003
License Upgrade Utility Complains That The Driver Is Not Installed 5/23/2003	
MediaAbort May Be Slow When Issued On Many Channels Simultaneously	5/23/2003
Web RLL Does Not Support Transfer Abort Or Timeouts	5/23/2003
WebRLL Race Condition Causes VOS To Crash	5/23/2003
REGID_LHTtsLoadUsrDict=3808 LHTTS Load User Dictionary Fails	5/23/2003
Attributes For MSI Conferencing Are Not Accessible	5/23/2003
Opening A Corrupt .WAV File Hangs Media Resource	5/23/2003
NetHub Plus Losing Messages Or Receiving Duplicates	5/23/2003
Caller ID Capability For DM3-Based MSI Stations Is Not Accessible	5/23/2003
Error gc_GetResourceH "Function Not Supported" Is Reported In Some Configurations	5/23/2003
db_first() Fails In VOS	5/23/2003
Unassign of L&H* TTS device takes 30 seconds to complete	5/21/2003
ws_SetTaskName Method In Sockets RLL Can Leak Memory	5/21/2003

\*Other names and brands may be claimed as the property of others.