Release Notes for Cisco Unified Call Services, Universal Edition 6.0(1) and Unified Call Studio 6.0(1)

November 2007

Content

Introduction

This document discusses new features, changes, and caveats for Release 6.0(1) of Cisco Unified Call Services, Universal Edition and Cisco Unified Call Studio software.

Additional information on new features, and on many of the product changes, are available in the relevant end-user documentation.

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Note: For the most up-to-date version of all Cisco documentation, go to the Cisco Web page: http://www.cisco.com/web/psa/products/index.html
In particular for the most up-to-date version of these release notes, go to the Cisco Web page:

http://www.cisco.com/en/US/products/ps7234/prod_release_notes_list.html and

http://www.cisco.com/en/US/products/ps7235/prod_release_notes_list.html

About Release 6.0(1)

Release 6.0(1) of Cisco Unified Call Services, Universal Edition and Unified Call Studio contains an extensive set of new functionality in Call Services and Call Studio, as well as an incremental set of defect fixes. Applying Release 6.0(1) installs all the functionality contained in Unified Call Services 3.6 and Call Studio 5.2, as well as the new 6.0(1) changes.

For changes that occurred prior to Release 6.0(1), refer to Appendix I.

The release is available on DVD and as downloadable installers from cisco.com.

A Note about Product Naming

These release notes reflect the following changes in the naming conventions within the Cisco Unified Communications product portfolio.

Audium Call Services is renamed to Cisco Unified Call Services, Universal Edition (abbreviated as Call Services).

Audium Studio is renamed to Cisco Unified Call Studio (abbreviated as Call Studio).

System Requirements

For hardware and third-party software specifications for Release 6.0(1), refer to the *Installation Guide for Cisco Unified Call Services, Universal Edition and Unified Call Studio, Release 6.0(1*).

Related Documentation

Documentation for Cisco Unified Call Services, Universal Edition, as well as most related documentation is accessible from

http://www.cisco.com/en/US/products/ps7234/tsd_products_support_series_home.html

Documentation for Cisco Unified Call Studio, as well as most related documentation is accessible from

http://www.cisco.com/en/US/products/ps7235/tsd_products_support_series_home.html

Release Notes for Cisco Unified Call Services, Universal Edition are accessible from

http://www.cisco.com/en/US/products/custcosw/ps7234/prod_release_notes_list.html

Release Notes for Cisco Unified Call Studio are accessible from

http://www.cisco.com/en/US/products/custcosw/ps7235/prod_release_notes_list.html

New and Changed Information

Release 6.0 contains new functionality and feature enhancements in Call Services and Call Studio, as well as a small set of defect fixes. All the functionality contained in Call Services 3.6 and Call Studio 5.2 is included in Release 6.0.

The following sections describe new features and enhancements that are pertinent to this release:

Web Services

Release 6.0 introduces native support for Web Services in Unified Call Services and Call Studio. A new call-flow element is included to support integration with sophisticated Web Services using Web Services Description Language (WSDL) and Service Oriented Architecture Protocol (SOAP) from directly within Call Studio and Call Services.

Voice Application Debugger

Release 6.0 provides the ability to execute and test voice application call flows directly from within Unified Call Studio.

Standalone Application Builder

Release 6.0 introduces the ability to deploy applications from Unified Call Studio to Call Services in unattended mode without running the graphical Call Studio.

Enhanced Multi-Language Support

Release 6.0 provides integrated and simplified support for configuring multilingual voice applications within Unified Call Studio.

Subroutine

Release 6.0 introduces a new Subdialog Invoke element in Unified Call Studio and Call Services to support the functionality of calling another voice application as a subdialog and passing data to and receive data from the subdialog application.

Application Management API

Release 6.0 provides the ability to monitor and configure Unified Call Services via any JMX-compliant (Java Management Extension) management interface.

Global Logger

Release 6.0 introduces the Global Logger in Unified Call Services to utilize Logging API and support full customization of global logs.

Eclipse Version Upgrade

Unified Call Studio in Release 6.0 is upgraded from Eclipse 3.1 to Eclipse 3.2.

Java Version Upgrade

Unified Call Services and Call Studio in Release 6.0 are upgraded from Java 4 to Java 5.

VoiceXML Gateway Description

Release 6.0 adds a new VoiceXML Gateway Description field to the “General Settings” properties pane in Unified Call Studio, providing additional information about a selected gateway adapter.

Local Hotlinks

In addition to retaining support for global hotlinks, Release 6.0 provides the ability to configure local hotlinks on a per-element basis in Unified Call Studio and Call Services.

Enhanced Security

Unified Call Services introduces security enhancement in most of the voice elements, by providing a new setting to disable logging of sensitive data containing callers' responses on a per-element basis.

Extended N-best Support

Release 6.0 extends support for n-best processing from Form elements to most of the voice elements in Unified Call Studio and Call Services.

Third Party Library Removal

The following third party libraries are no longer shipped with Unified Call Services and Call Studio as of Release 6.0:

- Xalan.jar
- Crimison.jar

If you have custom elements that require any of these libraries, , they can be downloaded from the internet.

audium.jar Library

The audium.jar library is no longer shipped with Unified Call Services as of Release 6.0. The framework.jar library should be used instead.

Discontinued Gateway Adapters

The following gateway adapters, which were supported in Call Service 3.6, are no longer supported as of Release 6.0:

- "VoiceGenie 7 with Nuance 8.5"

- "Avaya IR 1.3/2.0 with OSR 3.0"

Supported Gateway Adapters

The following gateway adapters are supported as of Release 6.0:

- "Cisco Unified CVP 4.1 with Cisco DTMF"

- "Cisco Unified CVP 4.1 with Nuance 8.5"

- "Cisco Unified CVP 4.1 with OSR 3/Nuance 9"

- "Cisco Unified CVP 4.1 VoiceXML 2.1 with Cisco DTMF"

- "Cisco Unified CVP 4.1 VoiceXML 2.1 with Nuance 8.5"

- "Cisco Unified CVP 4.1 VoiceXML 2.1 with OSR 3/Nuance 9"

- "Avaya AVP 3 with OSR"

- "Genesys GVP 7.2 with Genesys DTMF"

- "Genesys GVP 7.2 with OSR 3.0"
- "Intervoice EVIP 10 with OSR 3"
- "Syntellect 5.1 with Nuance 8.5"
- "Tellme with Nuance 8"
- "Tellme VoiceXML 2.1 with Nuance 8"
- "VoiceGenie 7 with OSR 3"

Installation Notes

Call Studio Operating System Support

No longer supported on Windows 2000, Unified Call Studio now provides debugging capability and should run on a developer workstation running either Windows XP or Windows Vista.

Call Studio Install and Vista "Program Files" Directory

Windows Vista sets access permissions on the C:\Program Files directory and allows only users with administrator privileges to write to this directory. Therefore, it is generally recommended that Call Studio be installed to a directory other than "C:\Program Files".

Call Services Install

The Call Services 6.0(1) installer only supports a full installation; there is no option to perform an update to an existing Call Services installation. It is highly recommended to back up all deployed voice applications as well as custom audio files manually in a temporary directory prior to installing Release 6.0(1).

WAR File Renamed

The WAR file for Call Services is renamed from Audium.war to CallServices.war as of Release 6.0(1).

EAR File on WebLogic

The Call Services 6.0(1) installs a WAR file on all supported application servers except for WebLogic. On WebLogic, an EAR file under the name “UnifiedCallServices.ear” will be installed instead.

Installation on WebSphere

There are manual steps that must be performed to finalize the Call Services installation on WebSphere. For example, the default port on WebSphere for web applications is 9080; it must be set to 7000 to work with Unified Call Services. See *Installation Guide for Cisco Unified Call Services, Universal Edition and Unified Call Studio, Release 6.0(1*) for additional information.

After the installation completes, install the WAR file via WebSphere’s standard web application deployment process. The WAR file is named CallServices.war; the file resides in C:\Program Files\IBM\WebSphere\AppServer\installableApps by default or a custom-specified location configured during installation When the CallServices.war is deployed, modify the module class loader order for CallServices.war using the WebSphere Administrative Console. To do this, in the WebSphere Administrative Console, navigate to Applications > Enterprise Applications > CallServices_war > Manage Modules > Cisco Unified Call Services, Universal Edition 6.0 and from the Class loader order drop-down, select Classes loaded with application class loader first. Click Apply, and then OK. Finally, save the changes to the master configuration. Refer to WebSphere documentation for additional details.

Custom JAR Files Locked on Windows

On the Microsoft Windows operating system, while Unified Call Services is running, a user attempting to delete an application folder by calling the “releaseApp” function may be prevented from doing so by the operating system if the application references any Java application archive (JAR) files placed within the java/application/lib or java/util/lib directories.

This is due to a known bug with Sun's JVM (see http:/bugs.sun.com/bugdatabase/view_bug.do?bug_id=5041014), where the system keeps an open file handle for JAR files not released until a garbage collection event occurs. As a result, the administrator will have to wait until the garbage collector activates before being able to delete the directory. The wait time is determined by how often garbage collection is run.

For the same reason, if a project is open in Unified Call Studio and it contains custom elements in its deploy/java folder, deleting the project may fail with the following dialog box message “Problems encountered while deleting files”. If this occurs, a workaround is to close Call Studio, and then reopen it to delete the remainder of the project

Call Studio Licensing

As of Release 6.0, Unified Call Studio can be used for 30 days after installation without an active license. This may be useful for simple testing or evaluation purposes. After 30 days, an active license must be applied to continue using Call Studio. Details on how to apply a license to Call Studio (and Call Services) are documented in the *Installation Guide for Cisco Unified Call Services, Universal Edition and Unified Call Studio, Release 6.0(1)*.

Note that a new license must be requested to work with Call Studio 6.0. Any license key issued for a previous version of Call Studio (Universal Edition) no longer works for this release.

Call Services Licensing

As of Release 6.0, Unified Call Services supports two concurrent sessions by default with no expiration date before a license is applied. To utilize more simultaneous sessions, Unified Call Services must be activated with a valid license.

Note that any active license key issued for a previous version of Call Services will continue to work for Release 6.0.

Important Notes

The following sections contain restrictions that apply to Release 6.0(1).

Subdialog Elements

The following five subdialog elements appear in Unified Call Studio as of Release 6.0: CVP Subdialog Start, CVP Subdialog Return, Subdialog Start, Subdialog Return, and Subdialog Invoke.

When using a Cisco Unified CVP VoiceXML gateway, CVP Subdialog Start and CVP Subdialog Return elements should be used. For a non-CVP VoiceXML gateway, Subdialog Start and Subdialog Return elements should be used instead.

SOAP Support in Web Service Element

Unified Call Services supports SOAP 1.1 in the Web Service element. Additionally, the WSDL file must not contain binding operations for SOAP 1.2 and HTTP POST.

Avoid Cyclic Type Definition in WSDL Schema

When a WSDL file's schema includes a cyclic type definition (that is, type A has child of type B, which has a child of type A), loading of the WSDL file into a Web Service element will fail and result in an error such as the following stored in Call Studio error log:

java.lang.StackOverflowError

at java.util.ArrayList.addAll(Unknown Source)

at org.apache.xmlbeans.impl.schema.SchemaTypeImpl.getProperties(SchemaTypeImpl.java:705)

... (etc.)

The cyclic type definition in the WSDL file's schema must be corrected for the WSDL file to load correctly.

Web Service Element on WebLogic 9.2

Unified Call Services throws SOAP message exceptions on WebLogic 9.2 when an application visits the Web Services element in the Callflow.

On WebLogic 9.2, Call Services is packaged as an enterprise application and shipped with all the libraries to provide Web Services support. However, WebLogic’s built-in Web Services packages are used to process SOAP messages if the application packages are not properly configured.

For Web Services to fully work on WebLogic 9.2, the following JVM option needs to be set:

-Djavax.xml.soap.MessageFactory=com.sun.xml.messaging.saaj.soap.ver1_1.SOAPMessageFactory1_1Im
p

This option enables WebLogic to use the SOAP message factory class provided by the enterprise application instead of its built-in packages. The option can be set in the setDomainEnv file among other script files.

Application Management API and WebLogic 9.2.2

When using a JMX-compliant client to connect to Call Services running on WebLogic 9.2.2, some Application Management operations (including the following) will fail and throw an exception:

- Suspend
- Deploy
- Resume
- Update application
- Update common classes

Note that the same issue does NOT occur on WebLogic 9.2.0, and the above issue appears to be caused by the updated XML API libraries shipped with WebLogic 9.2.2.

Subdialog Support on Multiple Application Servers

After voice applications are created to invoke other VoiceXML applications as subdialogs via the Subdialog Invoke element, Unified Call Services supports deployment of these applications to separate application servers.

If a calling application (for example, App 1) runs on a host having the same IP address as the subdialog application (for example, App 2), any of the sequences listed in (1) through (4) will work; however, those in (5) and (6) will fail due to cookie path conflicts between the servers:

1. App 1 on WebSphere calls App 2 on WebLogic (supported on same IP)
2. App 1 on WebSphere calls App 2 on Tomcat (supported on same IP)
3. App 1 on WebLogic calls App 2 on WebSphere (supported on same IP)
4. App 1 on WebLogic calls App 2 on Tomcat (supported on same IP)
5. App 1 on Tomcat calls App 2 on WebLogic (NOT supported on same IP)
6. App 1 on Tomcat calls App 2 on WebSphere (NOT supported on same IP)

If the calling application runs on a host having a different IP address from the subdialog application, then all six sequences will work as expected:

1. App 1 on WebSphere calls App 2 on WebLogic (supported on different IPs)
2. App 1 on WebSphere calls App 2 on Tomcat (supported on different IPs)
3. App 1 on WebLogic calls App 2 on WebSphere (supported on different IPs)
4. App 1 on WebLogic calls App 2 on Tomcat (supported on different IPs)
5. App 1 on Tomcat calls App 2 on WebLogic (supported on different IPs)
6. App 1 on Tomcat calls App 2 on WebSphere (supported on different IPs)

Limitations in Voice Application Debugger

The following restrictions apply to Voice Application Debugger in Unified Call Studio Release 6.0(1):

- Elements with external dependencies - Elements with external dependencies require special care when used with Voice Application Debugger; this includes both built-in and custom elements. Built-in elements that fall into this category include VoiceXML Insert, Subdialog Invoke, Application Transfer and Database. Since these elements require access to external systems which may not be accessible from the development machine they are being executed on, they may encounter errors at runtime. To avoid this, it is strongly recommended that such elements be skipped using the Skip and > context menu (available by right-clicking on the elements) option so that debugging can proceed past them. Once debugging is complete, they can be unskipped.
- Browser dependency – Custom elements that generate browser-specific VoiceXML markup (and therefore depend on a particular browser to work) may not function correctly. It is recommended that such elements be skipped using the Skip and > context menu option so that debugging can proceed past them. Once debugging is complete, they can be unskipped.
- Transfer element – All telephony transfer attempts, either via the built-in Transfer element or a custom transfer voice element, will be simulated as a blind transfer. When such an element is visited, the call flow will cease and the call end information will indicate that a transfer has occurred. This will occur even if the Transfer element is configured to perform a bridge transfer.
- External grammars - When a built-in or custom element using external grammars is visited, your input will be treated directly as the semantic interpretation. However, if there are any inline or built-in grammars active during that element, your input will first be matched against those grammars before being matched against external grammars.
- Input Mode - The following voice elements only accept DTMF input (regardless of **Input Mode** configuration): Currency, Currency_With_Confirm, Date, Date_With_Confirm, Digits, Digits_With_Confirm, Number, Number_With_Confirm, Phone, Phone_With_Confirm, Time and Time_With_Confirm.

- Element Data – Element data named “nbestLength” will always be returned as “1”, and “nbestInputmode” will always be returned as “voice”.

Limitations in Multi-Language Configuration

The following restrictions apply to multi-language configuration in Unified Call Studio Release 6.0(1):

- Global Audio – Application-level Audio Settings do not support multiple languages.
- Hotlink element – Hotlink grammars do not support multiple languages.

Limitations in Cisco CVP 4.1 Gateway Adapters

- Record and Record_With_Confirm elements – When **Start With Beep** is set to "true", no beep tone is played prior to recording. This restriction applies to the following Cisco CVP gateway adapters:

  - “Cisco Unified CVP 4.1 with Cisco DTMF”

  - “Cisco Unified CVP 4.1 with OSR 3/Nuance 9”

  - “Cisco Unified CVP 4.1 with Nuance 8.5”
- Transfer element - When a TTS error occurs during the blind transfer, multiple semantic errors are also returned. This is because the browser treats a blind transfer as a consultation transfer, and therefore monitors the outcome after the call transfer is attempted. To stop the semantic errors from occurring, in the application where a blind transfer is configured, add a Hotevent element with "Event" set to "error.com.cisco.media.resource.failure.tts" and the "Has Exit State" field checked. Then follow this Hotevent element with a Hang Up. This restriction applies to the following CVP gateway adapters:

  - “Cisco Unified CVP 4.1 with Cisco DTMF”

  - “Cisco Unified CVP 4.1 with OSR 3/Nuance 9”

  - “Cisco Unified CVP 4.1 with Nuance 8.5”
- TTS and Recorded Audios - Any audio item configured with only **Audio File URI** (and no **TTS**) will not be played if there is another audio item in the same Audio Group with only **TTS** configuration. This restriction applies to the following CVP gateway adapters:

  - “Cisco Unified CVP 4.1 with Nuance 8.5”

  - “Cisco Unified CVP 4.1 VoiceXML 2.1 with Nuance 8.5”
- Timeout properties - Configuration of the termtimeout property is not supported. Any configuration of the interdigittimeout property will dictate the terminating timeout behavior in DTMF input recognition. This restriction applies to the following CVP gateway adapters:

  - “Cisco Unified CVP 4.1 with Nuance 8.5”

  - “Cisco Unified CVP 4.1 VoiceXML 2.1 with Nuance 8.5”
- Custom elements - The following Cisco VoiceXML tag extension is not supported for custom elements: <cisco-vcrcontrol>. In order to use this extension, a VoiceXML Insert element should be used. This restriction applies to all CVP 4.1 gateway adapters.
- Asterisk * - The asterisk (star) key * is accepted as a valid input in Currency, Currency_With_Confirm, Digits and Digits_With_Confirm elements. This restriction applies to the following CVP gateway adapters:

  - “Cisco Unified CVP 4.1 with Cisco DTMF”

  - “Cisco Unified CVP 4.1 VoiceXML 2.1 with Cisco DTMF”

- Digits and Digits_With_Confirm elements – Depending on the input, the built-in digits grammar referenced by these elements may return a value in an incompatible format with the **Say It Smart** "Digit-by-Digit" type. It is therefore recommended that the result be post-processed before the "Digit-by-Digit" type is used. This restriction applies to the following CVP gateway adapters:

  - "Cisco Unified CVP 4.1 with Cisco DTMF"

  - "Cisco Unified CVP 4.1 VoiceXML 2.1 with Cisco DTMF"

- Phone and Phone_With_Confirm elements – Depending on the input, the built-in phone grammar referenced by these elements may return a value in an incompatible format with the **Say It Smart** "Phone Number" type. It is therefore recommended that the result be post-processed before the "Phone Number" type is used. This restriction applies to the following CVP gateway adapters:

  - "Cisco Unified CVP 4.1 with Cisco DTMF"

  - "Cisco Unified CVP 4.1 VoiceXML 2.1 with Cisco DTMF"

- Form and Form_With_Confirm elements - The asterisk (star) key * is a reserved character in Cisco DTMF grammar, and must be escaped to be included for recognition. To do this, set **DTMF Keypress** to "*" instead of "*".This restriction applies to the following CVP gateway adapters:

  - "Cisco Unified CVP 4.1 with Cisco DTMF"

  - "Cisco Unified CVP 4.1 VoiceXML 2.1 with Cisco DTMF"

- Speech recognition functionality is not available on the DTMF only adapters. Therefore, all voice elements with an **Input Mode** setting must have the setting configured to "dtmf". This restriction applies to the following CVP gateway adapters:

  - "Cisco Unified CVP 4.1 with Cisco DTMF"

  - "Cisco Unified CVP 4.1 VoiceXML 2.1 with Cisco DTMF"

- X_Option_Menu elements – Option X Voice settings (where X is 2 - 10 as applicable) must not be configured. This restriction applies to the following CVP gateway adapters:

  - "Cisco Unified CVP 4.1 with Cisco DTMF"

  - "Cisco Unified CVP 4.1 VoiceXML 2.1 with Cisco DTMF"

- Form and Form_With_Confirm elements – Cisco DTMF grammars can only be used inline, and therefore **DTMF Grammar** must NOT be configured. This restriction applies to the following CVP gateway adapters:

  - "Cisco Unified CVP 4.1 with Cisco DTMF"

  - "Cisco Unified CVP 4.1 VoiceXML 2.1 with Cisco DTMF"

- Form and Form_With_Confirm elements – Grammar slots and semantic interpretation are not supported by the Cisco DTMF grammar. Therefore, **Slot Element Data** should not be configured and **DTMF Keypress** should be configured without a return value. These restrictions apply to the following CVP gateway adapters:

  - "Cisco Unified CVP 4.1 with Cisco DTMF"

  - "Cisco Unified CVP 4.1 VoiceXML 2.1 with Cisco DTMF"

- Form and Form_With_Confirm elements – As the DTMF input process is terminated as soon as the browser detects a valid input, the developer should avoid setting **DTMF Keypress** to two inputs where one is a substring of the other. Otherwise, entering the longer input will always trigger a misrecognition error. For example, if **DTMF Keypress** is configured to accept DTMF inputs: 1 and 12, then a DTMF input 12 will be misrecognized as 1. This restriction applies to the following CVP gateway adapters:

  - "Cisco Unified CVP 4.1 with Cisco DTMF"

  - "Cisco Unified CVP 4.1 VoiceXML 2.1 with Cisco DTMF"

- Hotlink element – Speech recognition functionality is not available. Therefore, the **Speech** setting must not be configured. This restriction applies to the following CVP gateway adapters:

- “Cisco Unified CVP 4.1 with Cisco DTMF”

- “Cisco Unified CVP 4.1 VoiceXML 2.1 with Cisco DTMF”

- Hotlink element – External grammars are not supported. Therefore, the **DTMF** setting must not be configured with an External URI. This restriction applies to the following CVP gateway adapters:

  - “Cisco Unified CVP 4.1 with Cisco DTMF”

  - “Cisco Unified CVP 4.1 VoiceXML 2.1 with Cisco DTMF”

Limitations in Avaya AVP Gateway Adapter

- Phone and Phone_With_Confirm elements – Depending on the input, the built-in phone grammar referenced by these elements may return a value in an incompatible format with the **Say It Smart** “Phone Number” type. It is therefore recommended that the result be post-processed before the “Phone Number” type is used.
- Time and Time_With_Confirm elements – Depending on the input, the built-in phone grammar referenced by these elements may return a value in an incompatible format with the **Say It Smart** “Time/Time Period” type. It is therefore recommended that the result be post-processed before the “Time/Time Period” type is used.
- Form and Form_With_Confirm elements – Grammar slots are not supported. Therefore, **Slot Element Data** should not be configured.
- Transfer element – **Connect Timeout** is not supported during bridge transfer.

Limitations in Genesys GVP Gateway Adapters

- Record and Record_With_Confirm elements – When **Start With Beep** is set to "true", no beep tone is played prior to recording. This restriction applies to the following Genesys gateway adapters:

  - “Genesys GVP 7.2 with Genesys DTMF”

  - “Genesys GVP 7.2 with OSR 3.0”

- Record element – When **Terminate On DTMF** is set to "true", pressing a DTMF key will effectively terminate the recording; however, the same DTMF key is collected and buffered until the next input state, causing unexpected events (including nomatches and noinputs) to occur in that input state. In place of this element, a sequence of a Record element followed by an Audio element with **Barge In** set to “false” for the Initial Audio Group (to force deletion of the buffered DTMF input) should be used. This restriction applies to the following Genesys gateway adapter:

  - “Genesys GVP 7.2 with OSR 3.0”

- Record_With_Confirm element – When **Terminate On DTMF** is set to "true", pressing a DTMF key will effectively terminate the recording; however, the same DTMF key is collected and buffered until the confirmation state, causing unexpected events (including nomatches and noinputs) to occur. In place of this element, a sequence of a Record element followed by an Audio element with **Barge In** set to “false” for the Initial Audio Group and then a Yes_No_Menu element should be used. In order for the recorded audio to be played in the Yes_No_Menu, the recorded audio file must be saved in a HTTP accessible location. This restriction applies to the following Genesys gateway adapter:

  - “Genesys GVP 7.2 with OSR 3.0”

- Speech recognition functionality is not available. Therefore, all voice elements with an **Input Mode** setting must have the setting configured to "dtmf". This restriction applies to the following Genesys gateway adapter:

  - “Genesys GVP 7.2 with Genesys DTMF”

- X_Option_Menu elements – Option X Voice settings (where X is 2 - 10 as applicable) must not be configured. This restriction applies to the following Genesys gateway adapter:

  - “Genesys GVP 7.2 with Genesys DTMF”
- Form and Form_With_Confirm elements – External grammars are not supported. Therefore, **DTMF Grammar** must not be configured. This restriction applies to the following Genesys gateway adapter:

  - “Genesys GVP 7.2 with Genesys DTMF”
- Form and Form_With_Confirm elements – Grammar slots and semantic interpretation are not supported. Therefore, **Slot Element Data** should not be configured and **DTMF Keypress** should be configured without a return value. This restriction applies to the following Genesys gateway adapter:

  - “Genesys GVP 7.2 with Genesys DTMF”
- Hotlink element – Speech recognition functionality is not available. Therefore, the **Speech** setting must not be configured. This restriction applies to the following Genesys gateway adapter:

  - “Genesys GVP 7.2 with Genesys DTMF”
- Hotlink element – External grammars are not supported. Therefore, the **DTMF** setting must not be configured with an External URI. This restriction applies to the following Genesys gateway adapter:

  - “Genesys GVP 7.2 with Genesys DTMF”
- Custom elements – Genesys VoiceXML extensions that require the Telera namespace are not supported in custom elements. In order to use these extensions, the VoiceXML Insert element should be used instead, with the Telera namespace set to “http://www.telera.com/vxml/2.0/ext/20020430”. This restriction applies to the following Genesys gateway adapters:

  - “Genesys GVP 7.2 with Genesys DTMF”

  - “Genesys GVP 7.2 with OSR 3.0”

Limitations in Intervoice EVIP Gateway Adapter

- Form and Form_With_Confirm elements – The "maxnbest" setting is not supported for this adapter. Additionally, only one nbest result will be returned for these elements.
- Form and Form_With_Confirm elements – The interpretation string in Form element data starts with +JIT_literal:<interpretation_value> whenever slots are present. If the grammar does not contain slots, then this doesn't appear.
- Form and Form_With_Confirm elements – The inputmode element data will always have the value "dtmf", even when voice input is used. When "dtmf voice" and voice input is recognized, it reports application.lastresult$.inputmode as "dtmf".
- Form_With_Confirm elements – max noinput or max nomatch at the confirmation prompt will result in an error.semantic. In place of this element, a sequence of a Form element followed by a Yes_No_Menu element should be used.
- Record elements – The element must be configured to use "other" and MIME type "audio/x-wav".
- Record_With_Confirm elements – Voice input of "no" at confirmation prompt results in "error.semantic.ecmascript". In place of this element, a sequence of a Record element followed by a Yes_No_Menu element should be used. In order for the recorded audio to be played in the Yes_No_Menu, the recorded audio file must be saved in a HTTP accessible location.
- Hotevent Elements – The element’s **Event** setting cannot be configured with whitespace only, " ", or "."
- Hotlink Element – The element cannot be used if the application is invoking external VoiceXML via the VoiceXML Insert Element.
- Subdialog Invoke Element – The element is not supported in the case where the WebSphere application server is used and the subdialog applications are deployed to the same instance of Call Services.

- Use of alternate languages in the voice application is platform configuration dependent and functionality has not been confirmed.
- VoiceXML transfer through the EVIP browser is not currently supported.

Limitations in Syntellect Gateway Adapter

- Currency and Currency_With_Confirm elements – Depending on the input, the built-in phone grammar referenced by these elements may return a value in an incompatible format with the **Say It Smart** “Currency ($)” type. It is therefore recommended that the result be post-processed before the “Currency ($)” type is used.
- Phone and Phone_With_Confirm elements – Depending on the input, the built-in phone grammar referenced by these elements may return a value in an incompatible format with the **Say It Smart** "Phone Number" type. It is therefore recommended that the result be post-processed before the "Phone Number" type is used.
- Record and Record_With_Confirm elements are not supported on this adapter.
- Multilingual functionality is not supported on this adapter. While the action element named "Application_Modifier" can be used to change the language context at runtime, support for multiple concurrent languages per dialog state is not available on this adapter.

Limitations in Tellme Gateway Adapters

- Bargein cannot be disabled through the **Barge In** setting of individual audio groups. A partial workaround is to set a VoiceXML property named “bargein” to “false” for a voice element, which will disable bargein for all audio groups in that element. This restriction applies to the following Tellme gateway adapter:

  - “Tellme with Nuance 8”
- Hotlink element - External grammars are not supported. Therefore, the **DTMF** setting must not be configured with an External URI. This restriction applies to the following Tellme gateway adapter:

  - “Tellme with Nuance 8”
- Local and Global Hotlinks - When a voice application has a local hotlink configured to listen for the same DTMF input as a global hotlink, the global hotlink takes precedence over the local hotlink. This restriction applies to the following Tellme gateway adapter:

  - “Tellme VoiceXML 2.1 with Nuance 8”
- Whitespaces in inline DTMF grammars – The following inline DTMF grammar settings accepting multi-digit inputs should be configured without whitespaces: 1) the **DTMF Keypress** setting of Form and Form_With_Confirm elements, 2) the **DTMF** setting of local hotlinks, and 3) the **DTMF** setting of global hotlinks. This restriction applies to the following Tellme gateway adapters:

  - “Tellme with Nuance 8”

  - “Tellme VoiceXML 2.1 with Nuance 8”
- Phone and Phone_With_Confirm elements – Depending on the input, the built-in phone grammar referenced by these elements may return a value in an incompatible format with the **Say It Smart** “Phone Number” type. It is therefore recommended that the result be post-processed before the “Phone Number” type is used. This restriction applies to the following Tellme gateway adapters:

  - “Tellme with Nuance 8”

  - “Tellme VoiceXML 2.1 with Nuance 8”
- Time and Time_With_Confirm elements – Depending on the input, the built-in phone grammar referenced by these elements may return a value in an incompatible format with the **Say It Smart** “Time/Time Period”

type. It is therefore recommended that the result be post-processed before the “Time/Time Period” type is used. This restriction applies to the following Tellme gateway adapters:

- “Tellme with Nuance 8”

- “Tellme VoiceXML 2.1 with Nuance 8”

- Multilanguage functionality is not supported. While the action element named "Application_Modifier" can be used to change the language context at runtime, support for multiple concurrent languages per dialog state is not available. This restriction applies to the following Tellme gateway adapters:

  - “Tellme with Nuance 8”

  - “Tellme VoiceXML 2.1 with Nuance 8”

- Multilanguage functionality is not supported. While the action element named "Application_Modifier" can be used to change the language context at runtime, support for multiple concurrent languages per dialog state is not available. This restriction applies to the following Tellme gateway adapters:

  - “Tellme with Nuance 8”

  - “Tellme VoiceXML 2.1 with Nuance 8”

- Subdialog Invoke element – When a subdialog application is deployed locally on the same server as the invoking application, the call will transition properly from the calling application to the subdialog application, but may fail to return to the invoking application. As a workaround, refer to the VoiceXML probe page by appending a “&probe=true” at the end of the initial URI pointing to the FIRST application to be visited in the call (NOT the URI of the subdialog application). This restriction applies to the following Tellme gateway adapters:

  - “Tellme with Nuance 8”

  - “Tellme VoiceXML 2.1 with Nuance 8”

Limitations in VoiceGenie Gateway Adapter

- Currency and Currency_With_Confirm elements – Depending on the input, the built-in phone grammar referenced by these elements may return a value in an incompatible format with the **Say It Smart** “Currency ($)” type. It is therefore recommended that the result be post-processed before the “Currency ($)” type is used.
- Date and Date_With_Confirm elements – Depending on the input, the built-in phone grammar referenced by these elements may return a value in an incompatible format with the **Say It Smart** “Date” type. It is therefore recommended that the result be post-processed before the “Date” type is used.
- Phone and Phone_With_Confirm elements – Depending on the input, the built-in phone grammar referenced by these elements may return a value in an incompatible format with the **Say It Smart** “Phone Number” type. It is therefore recommended that the result be post-processed before the “Phone Number” type is used.
- Time and Time_With_Confirm elements – Depending on the input, the built-in phone grammar referenced by these elements may return a value in an incompatible format with the **Say It Smart** “Time/Time Period” type. It is therefore recommended that the result be post-processed before the “Time/Time Period” type is used.
- Form and Form_With_Confirm elements – When **Modal** is set to “true”, application-level universal grammars are not disabled by the browser. As a workaround, set the VoiceXML property “universals” to “none” from the “Settings” tab of the Element Configuration pane in Call Studio, to disable all application-level universal grammars.

Resolved Caveats in This Release

Resolved caveats are no longer listed in Release Notes. Instead you can find the latest resolved caveat information through Bug Toolkit, which is an online tool that is available for customers to query defects according to their own needs.

Tip You need an account with Cisco.com (Cisco Connection Online) to use the Bug Toolkit to find open and resolved caveats of any severity for any release.

To access the Bug Toolkit, log onto
http://www.cisco.com/cgi-bin/Support/Bugtool/launch_bugtool.pl

Bug Toolkit

To access Bug Toolkit, you need the following items:

- Internet connection
- Web browser
- Cisco.com user ID and password

Procedure

Tip To access the Bug Toolkit, go to
http://www.cisco.com/cgi-bin/Support/Bugtool/launch_bugtool.pl

Step 1 Log on with your Cisco.com user ID and password.

Step 2 Click the **Launch Bug Toolkit** hyperlink.

Step 3 If you are looking for information about a specific caveat, enter the ID number in the "Enter known bug ID:" field.

Step 4 Click **Next**. The Cisco Unified Intelligent Contact Management Enterprise search window displays.

Step 4

Step 5 Choose the filters to query for caveats. You can choose any or all of the available options:

a. Select the Cisco Unified Intelligent Contact Management Enterprise Version:

- Choose the major version for the major releases.

A major release contains significant new features, enhancements, architectural changes, and/or defect fixes.

- Choose the revision for more specific information.

A revision release primarily contains defect fixes to address specific problems, but it may also include new features and/or enhancements.

b. Choose the Features or Components to query; make your selection from the "Available" list and click **Add** to place your selection in the "Limit search to" list.

To query for all caveats for a specified release, choose "All Features" in the left window pane.

Note The default value specifies "All Features" and includes all of the items in the left window pane.

c.Enter keywords to search for a caveat title and description, if desired.

Note To make queries less specific, use the All wildcard for the major version/revision, features/components, and keyword options.

d. Choose the Set Advanced Options, including the following items:

- Bug Severity level—The default specifies 1-3.
- Bug Status Group—Check the Fixed check box for resolved caveats.
- Release Note Enclosure—The default specifies Valid Release Note Enclosure.

e. Click **Next**.

Step 6 Bug Toolkit returns the list of caveats on the basis of your query. You can modify your results by submitting another query and using different criteria.

Open Caveats in This Release

This section contains a list of defects that are currently pending in Cisco Unified Call Services 6.0(1) and Unified Call Studio 6.0(1). Defects are listed by component and then by identifier.

Tip If you have an account with Cisco.com, you can use the Bug Toolkit to find caveats of any severity for any release. Bug Toolkit may also provide a more current listing than is reflected in this document. To access the Bug Toolkit, log onto http://www.cisco.com/cgi-bin/Support/Bugtool/launch_bugtool.pl

Table 1 Open Caveats for Cisco Unified Call Services and Call Studio Release 6.0(1)

Identifier	Component	Headline
CSCsk45143	vxml_studio	Issues with debugger and VoiceXML inserts
CSCzc20983	vxml_server	On-hold call is not hearing busy message immediately

Obtaining Documentation, Obtaining Support, and Security Guidelines

For information on obtaining documentation, obtaining support, providing documentation feedback, security guidelines, and also recommended aliases and general Cisco documents, see the monthly *What's New in Cisco Product Documentation*, which also lists all new and revised Cisco technical documentation, at:

http://www.cisco.com/en/US/docs/general/whatsnew/whatsnew.html

Release Notes for Cisco Unified Call Services, Universal Edition 6.0(1) and Unified Call Studio 6.0(1)

Appendix I – Change Logs (Prior to This Release)

Changes in Call Studio (between 5.0 and 5.1)

Issue Key	Component	Summary
AS-349	Documenter	Documenter truncates application call flow images.
AS-531	Documenter	Documenter generates incomplete document for large applications.
AS-543	Views	java.lang.StackOverflowError thrown when creating/modifying Element java code in Audium Studio.
AS-546	Documenter	Documenter fails when element configuration data is incomplete.
AS-549	Decision Editor	Decision Editor resets "Operator" value when conditional operators are used.
AS-617	Other	Update Audium Studio to include libraries from Call Services 3.5 GA release.
AS-643	Decision Editor	The 2nd time it is opened, Decision Editor displays error dialog when row is selected.
AS-644	Decision Editor	Defining multiple conditions for the same exit state using Decision Editor creates several exit states with the same name.
AS-647	Views	Elements which use a pop-up text editor (Database, Email, etc.) experience setting corruption.
AS-664	Decision Editor	Sort items in the "Element Name" drop down box within Decision Editor.
AS-808	Licensing	Licensing client updates including fix for invalid system id generated on laptops
AS-277 AS-606 AS-607	Studio Preferences	Studio Preferences: Expand elements in Elements View, Expand call flow elements in Outline View and Call Flow Themes.
AS-585 AS-673	Application Settings	Interface to configure new Call Services 3.6 application settings including new loggers, application start/end and JavaScript in the root document.
AS-674	Application Settings / Elements	Interface for configuring an Audium voice application to be called as a subdialog, including new Subdialog Start and Subdialog Return elements.
AS-626	Application Settings / Deployment	Interface to support creating and deploying applications on multiple versions of Call Services.
AS-742	Deployment	New option to create an archive package of deployed application files.
AS-276	Callflow Editor	Navigating from Page Entry to Page Connectors.
AS-608 AS-745	Callflow Editor	Element comments including visual indication for elements which contain a comment.
AS-692	Decisions	Support new call data fields in Decision Editor.
AS-676	Substitution	Support new call data fields in Substitution drop down box.

AS-612 AS-615	Extension Kit	First implementation of the Audium Studio Extension Kit.
AS-157	Project Wizards	Automatic opening of a new/imported Audium project's application in the Callflow Editor.
AS-425	Element Configuration	Allow users to edit audio item comments via right-click context menu.
AS-581	Element Configuration	Allow users to rename audio items via right-click context menu.
AS-523	Decisions	Improved UI behavior when data type selected is reselected but not changed to a different value.
AS-700 AS-747	Documenter	Documenter wizard improvements including additional configuration options, breakdown of document sections and UI improvements.
AS-586 AS-587 AS-621 AS-622 AS-623 AS-650 AS-730 AS-753	Documenter	Document improvements including cleaner formatting, links between element configurations and content for new features.
AS-597 AS-693	Documenter	Memory and document file size improvements.
AS-604	Substitution	List known element data values as options in drop down box for Substitution.
AS-558 AS-737	Documentation	Studio documentation updates including updated screenshots and content for new features.
AS-167 AS-385 AS-435 AS-688	Images	Updated images in Studio (splash screen, deploy wizard, validate and deploy, Audium Builder perspective).
AS-641	UI	Updated grammar for text content in import dialog.

Changes in Call Studio (between 5.1 and 5.2)

Issue Key	**Component**	**Summary**
AS-790	Documentation	Update Studio documentation for subdialog elements
AS-774	Licensing	Add a URL to the manual activation page in the activation dialog
AS-796	Application Settings	Update default activity logger configuration to include error events for all logging levels
AS-802	Application Settings/Logging	Importing apps which were developed with previous Studio versions does not add custom loggers
AS-801	Application Settings/Logging	Default logger naming consistency
AS-797	Callflow Editor	Renaming an element to change capitalization in Studio causes that element to lose all other future changes

AS-808	Licensing	Licensing client updates including fix for invalid system id generated on laptops
AS-814	Deployment	Remote deployment may fail due to the Audium Home path always being updated to use the Windows system file separator
AS-797	Callflow Editor	Renaming an element to change capitalization in Studio causes that element to lose all other future changes
AS-805 AS-806 AS-815 AS-816	Documentation	Audium Studio Online documentation updates including updated content, screen shots and search results
AS-808	Licensing	Licensing client updates including fix for invalid system id generated on laptops
AS-814	Deployment	Remote deployment may fail due to the Audium Home path always being updated to use the Windows system file separator
AS-823	Application Settings/Logging	Default logger configuration updates
AS-824	Documentation	Audium Studio Online documentation sync up with hard copy including updated content and formatting.
AS-828	Application Settings/Logging	Default activity logger configuration updates to rotate by size.
AS-833	Deployment	App-specific OEM admin scripts updated for deployed applications.
AS-835	Documentation	Audium Studio Online documentation updates including updated content and screen shots.
CSCsf99414	Deployment	Archive deployment update to fix issue with executable permissions of admin scripts on Linux, AIX, etc.
CSCsg49066	Substitution	NullPointerException thrown when opening Substitution Tag Builder dialog.
CSCzc10353	Element Configuration / Extension Points	Added extension points to extend Static / Say It Smart Audio Item configurations.
CSCzc10526	Decision Editor	Added AND/OR support for expressions in the Decision Editor.
CSCzc10527	Element Groups	Added the ability to ungroup elements inside an Element Group.
CSCzc10090	Deployment	Updated deploy dialog to display error message when trying to deploy an unsaved application.
CSCzc10086 CSCzc10192 CSCzc10221 CSCzc10783	Element Configuration	Updated Element Configuration View to include scrollbars when needed.
CSCzc10232	Element Configuration	Updated textfield setting dialog to include Substitution button.
CSCzc10436	Prompt Manager	Fixed issue where applying column selection removes the column sort image.
CSCzc10525	Decision Editor	List known exit state values as options in a drop down box.
CSCzc10746 CSCzc10754	Decision Editor	List known element data / session data values as options in drop down boxes.
CSCzc10760	Substitution	List known session data values as options in a drop down box.
CSCzc10776	Documentation	Online documentation updates including updated content and screen shots.

CSCzc10778	Studio Preferences / Application Settings	List configurable language / encoding values as options in drop down boxes.
CSCzc10780	Callflow Editor	Updated context menu to include alignment tools.
CSCsh17447	Elements	Updated elements.jar to fix issue with incorrect Element Data drop down options.
CSCsg94814 CSCsh07115	Libraries	Third party packaging updates.

Changes in Call Services (between 3.5 and 3.6.10)

Issue Key	Summary
AC-23, AC-90, AC-377, AC-69, AC-91	Re-architected how logging works on Call Services to use individual "loggers" that plug into the system and perform logging in any way desired. Developers can now create custom loggers by using a new logger API. Loggers are deployed on Call Services in the same way elements are, and can refer to configurations to allow for detailed modification of behavior. Call Services now includes four default loggers that produce the same logs as prior Call Services releases but with significant new customizability. The activity logger produces the activity log and can be configured to customize the level of detail reported, control the log file formats, rotate logs in one of four different ways, use different kinds of memory caches, and purge old log files using several different strategies. The error logger produces the error log and can be configured to control the log file format and purge old log files using several different strategies. The admin logger produces the administration log. The debug logger produces a record of each HTTP request and response made for a call to an application, including the VoiceXML sent back to the voice browser.
AC-413	Introduced two new ways to create and maintain variable information: application data and global data. Application data variables are shared across all calls to an application and global data variables are shared across all calls to all applications on the system. To provide additional access to these variables, two new component types were introduced: start of application and end of application classes that are called when an application starts or ends. The Java API was updated to allow for the construction of these two new components.
AC-354, AC-396, AC-472, AC-487	Implemented a new licensing scheme for Call Services to make the process of authorizing Call Services easier, faster, more flexible, and more in line with the licensing mechanism for other Audium products. Call Services is now licensed through the use of a web console that will automatically download appropriate license files once the user has a valid account with Audium. A manual license activation process is also available should the system not have direct outside access. Licenses can be automatically reissued when new hardware is used, no longer requiring direct contact with Audium. Additionally, Call Services licenses are no longer tied to the IP address of the system and uses a more robust and accurate mechanism to tie a license to a physical machine.
AC-183	Audium Call Services is now compiled in Java 1.4. ***As a result, all custom components written in Java that are deployed on Call Services such as elements Say It Smart plugins, and application-related code such as action and decision elements, must now be compiled in Java 1.4 in order to be supported by Audium***. In most cases, all that is required is a simple recompilation.
AC-338	Call Services now supports being run on JDK 1.5 on the supported application servers that allow it (such as Tomcat 5.5). Note that Audium will support only *running* Call Services on JDK 1.5, not compiling custom code in JDK 1.5 (see AC-183).
AC-376	Improved multi-language support. There is now a single language and encoding application setting, language and encoding are now available via substitution, different audio groups can now be assigned to different languages, and the language and encoding can now be changed without the need for programming.
AE-77	Introduced a new action element named "Application_Modifier" that provides the ability for the application designer to perform tasks previously only possible by writing code. The element can now change the application maintainer, language, encoding and default audio path. It also provides a mechanism to delete session data.

AE-76	Introduced a new action element named "Math" that provides the ability for the application designer to execute mathematical equations previously only possible by writing code. The equations can involve numeric arithmetic, boolean arithmetic, trigonometry and other mathematical functions.
AC-353	Call Services now fully supports any Audium application being treated as a VoiceXML subdialog by simply changing an application setting.
AC-601	During the application startup process, Call Services will automatically convert the XML files of any application originally deployed for the 3.5 version of Call Services.
AC-582, AC-590	Improved the mechanism for ending sessions, giving components that run at the end of a call the ability to access the session during a certain period of time. This period of time is configurable via the Call Services global configuration. Additionally, administration script outputs will now display active callers (the number of callers actively visiting the application) and sessions ending (the sessions that are waiting for the aforementioned period of time before ending).
AC-578	Updated the Date Say It Smart plugin to support inputs using 1-digit values for a component of the date. This allows for the use of substitution to define the input for playing back the date.
AC-61	Improved the performance of the system with regards to how threads work for logging. The administrator can now control the thread usage devoted to logging.
AC-158	Introduced a web console in the Call Services web application that reports various information about the installation such as the Java version referenced, Call Services version and licensing information, gateway adapters installed, etc.
AC-214	Custom JavaScript functions can now be added by an application designer to the root document of an application. This can be done manually through the application settings or programmatically.
AC-451	Simplified the process of downloading and installing gateway adapters when updates are available to Call Services or adapters.
AC-200, AC-382, AC-318	Updated the Call Services global configuration to allow for the changing of the HTTP port on which administration scripts communicate with Call Services.
AC-404	Added support for IBM WebSphere 5.1 running on Microsoft Windows 2003 Server.
AC-686	Updated the `VTransfer` VFC to accept a maximum transfer time of 0, which according to the VoiceXML 2.0 specification will set an unlimited duration. Previously, `VTransfer` threw an exception in this case.
AE-68	The Database element's "Session Data Key" setting is now required.
AC-626	The Log4j version included in the Call Services web application archive has been upgraded to version 1.2.12.
AC-679	The HelloWorld application included with Call Services has been changed to reference the Tellme gateway adapter.
AC-500	The script that provided the ability to import activity log files to databases has been eliminated.
AC-410	Improved Audium voice elements to eliminate the use of an extra VoiceXML variable named "the_main_field_name".
AC-197	Fixed a bug that prevented the Call Services web application from being updated or reloaded by the application server, complaining about another instance of Call Services running.
AC-323, AC-526, AC-674	Fixed bugs with the way calls were being managed when put on hold that would activate the start of call action prematurely and fail to apply the session timeout to on hold sessions.
AC-378, AC-607	Fixed various bugs with the Call Services SNMP functionality.
AC-399, AC-465	Resolved conflicts caused by the third party Java library JDOM.

AC-709	Fixed bugs that could cause boolean values appearing in an application's settings to load incorrectly.
AC-333	Fixed a small bug that added a trailing slash to the return of the APIBase class methods getAudiumHome and getApplicationDirectory even if the AUDIUM_HOME environment variable already had a trailing slash.
AE-63	Updated a tooltip for the Record_With_Confirm element for clarification.
AC-223, AC-622	Updated various error messages for clarification.
AC-594, AC-787	Added support for running Call Services on Websphere 5.1 and 6.1 on the AIX operating system.
AC-832	Improved User Guide to include more information on the included loggers.
AC-559, AC-733	Updated various licensing messages for clarity.
AC-637	Updated the internal mechanism Call Services uses to evaluate XML decisions to eliminate the Mandarax rules engine.
AC-746	Fixed a bug that caused the getName method of VoiceElementConfig.AudioItem to always return null when accessed within a dynamic configuration Java class.
AC-752	Fixed a bug that in rare situations could cause a NullPointerException to appear on the application server console. This bug could also cause issues when Call Services is run in a session replication environment.
AC-761	Resolved several misspellings in error messages.
AC-675	Fixed a bug introduced in the 3.6 release that prevented a dynamic configuration Java class from altering the base configuration object passed to it. This bug did not exist if no base configuration existed.
AC-775	Fixed a bug that neglected to include the new call data types "language" and "encoding" in XML decisions. These new call data types were introduced in the 3.6 release.
AC-776	Fixed a bug that caused the <called_from_ani> tag in XML decisions to not return the correct value.
AC-778, AC-781, AC-812	Fixed several incorrect pieces of information in the User Guide when describing XML decisions.
AC-780	Fixed various bugs that existed when using the <historical_data> tag in XML decisions.
AC-785	Fixed an omission in the User Guide to include 4 error events as events that the activity logger can listen to.
AC-786	Updated the Call Services feature that automatically converts old applications, to include error events in the default activity logger configuration.
AC-788	Fixed a bug where a VoiceXML error would cause the element exit event to be thrown twice, which was causing loggers listening to this event to be activated twice.
AC-798	Fixed a bug in the activity logger that would incorrectly put the element name in the category column when an error occurred.
AC-799	Fixed a misspelling of two constants defined in the logger interface class IApplicationEventIDs: APPLICATON_ADMINISTRATION_EVENT_ID and APPLICATON_ADMINISTRATION_ERROR_EVENT_ID.
LIC-136, AC-801	The licensing system was updated to remove a dependency on a third party library for Microsoft Windows system ID generation. This eliminated the need for the Fingerprint.dll library files appearing under the Call Services lib directory.
AC-807	Updated a misspelling in the VIfGroup Javadocs.
AC-839	Fixed a bug that caused an exception to be thrown when Call Services attempted to handle a perpetual license (one that does not have an expiration date).
AC-814	Fixed a bug that would incorrectly create session data variables should the values session.connection.remote.uri

	or session.connection.local.uri be sent to the initial URL for a call, potentially causing issues when those variables are referenced in VoiceXML Insert elements.
AC-813	Fixed a rare bug that would have displayed negative values in the Call Services global status script output when Call Services runs on application servers that perform session persistence between application server restarts (such as Apache Tomcat 5.x).
AC-811	Fixed a bug in the license page caused by entering an installation key then pressing the "Back" button on the web browser and then entering a different installation key. The new installation key is now no longer ignored.
AC-810, AC-816	Fixed a bug that caused ConcurrentModificationExceptions to appear when loggers access the Session API methods "getAllElementData", "renameElementData", "removeAllElementData", "getAllSessionData", "removeAllSessionData", "getAllLoggerScratchData", "removeLoggerScratchData", or "removeAllLoggerScratchData". Note that even with the fix, the methods "getAllSessionData" and "getAllLoggerScratchData" could still encounter these exceptions unless the developer synchronizes on the HashMap object returned by these methods.
AC-802	Fixed a small bug that could have caused a session in a session replication environment to fail to replicate correctly.
AC-817	Updated licensing to address issues on Windows that could cause System IDs to change despite being used on the same computer.
AC-860, AC-869, CSCzc20869	Added support for Call Services to run on the BEA Weblogic 9.2 Java application server.
AC-867	For certain OEM versions of Call Services, the default Activity Logger rotation strategy has been changed to rotate by size, with a limit of 100MB.
AC-870	The Intervoice gateway adapter has been updated to support specifying a grammar weight and the proprietary attribute "srcexpr" of the VoiceXML `<grammar>` tag for hotlinks.
AC-812, CSCzc20812	Fixed a bug in the User Guide that incorrectly listed the "n" attribute of the `<nth_element>` and `<nth_exit_state>` XML decision tags as requiring a 0-based number when in fact it requires a 1-based number (meaning the first element in the callflow has an index of 1, not 0).
AC-839, CSCzc20839	Fixed an issue that would cause the Software Activation Console to encounter an error when a specific type of license was being issued.
AC-851, CSCzc20851	Fixed a bug that returned a null name for a Say It Smart audio item when accessed from within a dynamic configuration Java class using the `getName` method of the `VoiceElementConfig.AudioItem` class.
AC-855, CSCzc20855	Fixed bugs in the Activity Logger that would cause exceptions to be thrown under load when rotating by hour or by size.
AC-861, CSCzc20861	Fixed various bugs that prevented the file purging capabilities of the Activity Logger to function correctly.
AC-862, CSCzc20862	Fixed bugs that made the Admin Logger, Error Logger, and Debug Logger to not use the appropriate operating system specific carriage returns for the logs. This had the affect on Microsoft Windows of making the files produced by these loggers very difficult to read when opened with Notepad.
AC-873, CSCzc20873	Fixed bugs that caused exceptions to appear when attempting to access the Software Activation Console, the information page, or call into an application if the AUDIUM_HOME value is not set. The fix involved producing user-readable web or VoiceXML pages in these situations.
AC-874, CSCzc20874	Fixed bugs that caused exceptions to appear when attempting to access the information page or call into an application when Call Services is not licensed. The fix involved producing user-readable web or VoiceXML pages in these situations.
AC-885	Fixed an issue appearing in OEM versions of the included HelloWorld application not being built with the correct version of Studio.
AC-881,	Fixed a bug that prevented element data created in a configurable decision element from triggering an

CSCzc20881	`ElementDataEvent` logger event when the `setElementData(String variableName,String value,int type, boolean log)` method were called with a "true" as the last parameter. The result of this bug was that element data created in configurable decision elements was not logged in the activity log or any other logger listening for this event.
AE-111	Fixed a minor naming inconsistency of a VoiceXML field produced by the Time and Time_With_Confirm elements
AE-65	Removed an unnecessary Javascript variable declaration appearing in the VoiceXML pages generated by the Form and Form_With_Confirm elements.
AE-60	Updated the `VBuiltInField` Voice Foundation Class so that when performing a boolean capture, the "type" VoiceXML attribute does not explicitly appear unless the developer maps the yes and no choices to DTMF values other than 1 and 2, respectively.
CSCsg21062	Added a new method to com.audium.server.session.ComponentAPI to return a shallow clone of the session data HashMap. This was done to allow for more flexible options for accessing all session data.
CSCsg40157	All threads used by Call Services are now given descriptive names to better able to track Call Services specific threads when reading a thread dump. Logger threads are given unique names to provide clues as to which logger is handling which event for which session.
AC-839, CSCzc20839	Fixed a bug in the Software Activation Console that caused an error to occur when a permanent license was granted.
CSCsg39535	Fixed a small bug that neglected to include the name of the VoiceXML event in the activity log when the gateway encountered a VoiceXML event.
CSCsg12261	Fixed a bug in the application error logger that would print a stack trace to the application server console. The error is now only reported in an error log.
CSCsg21268	Fixed a bug that caused an error to appear in the global error log at the end of a call in progress when an application was updated.
CSCsg25096	Fixed a bug that caused the active session count outputted by the global or application-specific status script to display 0 calls when an application is updated while calls to the application were in progress.
CSCsg23484	Fixed a bug that displayed an error in the global error log when running the application-specific status script after the application was updated while calls to the application were in progress.
CSCsg23901	Fixed a bug that caused an exception to be thrown when the deployment of an application failed.
CSCsg23646	Fixed a bug that caused an error message to incorrectly appear in the global error log when the deployAllNewApps script was executed.
CSCsg04689	Fixed a bug that failed to increment the active session counter when a call that was on hold enters an application.
CSCsg46199	General improvements in error handling encountered when applications are loaded.
CSCsg34250	Fixed a bug that did not run the application settings converter when an application is updated with a pre-3.6.x release version of the application and the update administration script, causing errors to appear.
CSCsg50411	Fixed a bug that could cause an exception to be thrown when the global status script was accessed when the system was under load.
CSCsg54137	Fixed a bug in the XML API that did not include the <call_ended> tag in the "inputs" XML document sent for the on call end event.
CSCsg53941	Fixed a bug that could cause a concurrency error when calling the Session API method `getElementDataType` while iterating through the `HashMap` returned by the Session API method `getAllElementData`.
CSCsg12470	Made code changes both for handling exceptions that occur with application loggers and for managing the event queue. Global tuning parameters `poll_interval_ms` and `events_per_iteration` now default to 10ms and 5000 respectively.

CSCsg15677	Fixed a bug to ensure that the proper element data is reported by the following elements (for use in various Audium Studio drop-down menus): `MBasicCurrency, MBasicDate, MBasicDigit, MFoundationForm, MFoundationFormWithConfirm, MBasicNumber, MBasicPhone, MBasicTime`. The common change was that `confidence` was changed to `value_confidence`, but the Form elements also had `nbestLength` added to match with the Element Specification document.
CSCsh16431	Fixed a bug that prevented the built-in Date, Phone and Time elements from storing "value" element data at runtime.

Known Issues in Call Services 3.6.10

Issue Key	**Summary**
CSCzc20688	On certain application server / voice browser combinations, a single phone call that involves more than one Audium application as subdialogs will likely cause issues within the second application when the time the call spends between applications is less than the session invalidation delay. This can be minimized by reducing the session invalidation delay, however if the delay is set to 0, loggers that listen for the end of call event (including the activity logger) will encounter errors. This behavior is highly dependent on the application server and voice browser combination as some combinations do not have this issue while others do. Testing would have to be done on a combination to determine if there is an issue. It is expected that this known issue would affect only a tiny number of implementations. Note that this known issue is resolved in Release 6.0(1).
CSCzc20606	Identified a known issue involving the logging of the activation of Hotlinks and Hotevents that do not have exit states. Due to the nature of VoiceXML, when a Hotlink or Hotevent that does not have an exit state is activated, Audium Call Services will not be notified of that fact and hence will not cause any logger events to occur. As a result, the activation will not be noted in the activity log for that application, nor any logger listening for those logger events. Note that this known issue is not resolved in Release 6.0(1).
CSCzc20303	A substitution value within a voice element configuration's VoiceXML property referring to data that does not exist will cause a warning message to appear in the activity log multiple times for most voice elements. The more VoiceXML pages a voice element produces, the more times this warning message will appear in the activity log for that element. Note that this known issue is not resolved in Release 6.0(1).
CSCzc20211	Due to a bug in the Sun Java Virtual Machine in the way classloaders work with JAR files, when releasing an application on Microsoft Windows that contains custom JAR files in its `java` folder, those JAR files will remain open thereby preventing the immediate deletion of the application folder. However, as soon as incremental garbage collection occurs, the JAR files will be closed, allowing the application folder to be deleted. The frequency with which incremental garbage collection runs depends on the call volume handled by the system. Note that this known issue is not resolved in Release 6.0(1).
CSCzc20134	If a substitution within an application transfer element refers to data that does not exist, a warning message will appear in the activity log of the element that appears previous to the application transfer element in the call flow, making it look like the substitution occurred within that element. Note that this known issue is resolved in Release 6.0(1).