Migrating Legacy IVR Applications to VoiceXML with Voxeo

The advantages of a 100% VoiceXML compliant platform
# TABLE OF CONTENTS

1 Introduction .......................................................................................................................... 2
2 About Voxeo CXP ................................................................................................................ 2
3 Speeding Migration to VoiceXML with Voxeo ................................................................. 4
   Development Productivity ................................................................................................. 4
   CXPML ............................................................................................................................ 5
   Application Code ............................................................................................................. 9
   Other Resources ............................................................................................................. 9
   Migration Tools and Partners ....................................................................................... 9
4 Migration Lessons Learned ............................................................................................... 10
5 Voxeo Unlocked Communications .................................................................................. 11
6 Learn More ..................................................................................................................... 12
INTRODUCTION

Many enterprises and voice application providers are facing IVR platform end-of-life. With most vendors offering dialog design tools that lock customers into their own platform, platform end-of-life usually means application end-of-life too. VoiceXML is a mature and well-adopted standard in the telephony and IVR industry — and the primary target format for voice application migration today. But what is the best way to move your proprietary legacy applications to VoiceXML? And more importantly, how do you ensure you won’t be locked in again?

Voxeo can help you:

- **Speed migration to VoiceXML, as well as the migration of hand-coded VoiceXML to a superior VoiceXML-based application environment**
- **Develop applications using a platform-independent service creation environment that protects your investment with automatic portability to over 30 leading IVR platforms, including but not limited to Voxeo’s own Prophecy IVR**
- **Avoid the lock-in pitfall by using a platform that is 100% standards compliant and unlocked at every layer — enabling your solution to easily evolve with the needs of your business**
- **Leverage your new platform post-migration and use the power of Unlocked Communications to provide sophisticated, multi-channel self service options and a unified view of your customers**
- **Access a large community of developers and certified partners equally committed to standards-based development, “future-proofing” your new system so that it can evolve along with customer contact preferences and your business needs**

Voxeo CXP provides service creation, execution, management, reporting and analytics capabilities that are unmatched in the industry. While companies understand the value of having a comprehensive, mature product for their voice application endeavors, they fear being locked in to a single vendor. Unlike other tools, CXP provides an open and flexible environment that eliminates vendor lock-in at every level. We’ll explain how you can use this powerful tool to simplify the migration of your existing applications and discuss how both CXP and the Voxeo Prophecy IVR platform are designed to protect your investment and put you in control.

ABOUT CXP

Voxeo CXP, formerly known as VoiceObjects, is a carrier-grade Application Lifecycle suite for Unified Self-Service. The solution is based on open standards such as VoiceXML, HTTP, (X)HTML, XML, SQL, and SIP. Using CXP, developers can deploy one service definition to interact with customers over multiple channels:

- **Voice**
- **Mobile Web**
- **Short Message Service (SMS)**
- **Unstructured Supplementary Services Data (USSD)**
- **Social Networking (Twitter monitoring and response)**
- **Smartphone Apps**

This is realized through CXP’s Layer concept, which allows the creation and maintenance of services that share application logic and backend connectors, but have different presentation layers. The same concept is used to support highly personalized applications, such as multi-lingual self-service or applications that adapt to caller preferences, transactions, speaking styles, etc.
CXP Developer, the graphical service creation and management environment, comes as a Web-based thin-client front-end as well as an Eclipse-based rich client plug-in to suit both administrator and developer needs. The Web front-end provides a Control Center for quick access to server and service deployment statuses, log files, trace files, system configuration, etc. The Eclipse-based development environment offers embedded features such as drag and drop, auto-completion, testing and debugging tools, automatic documentation, and much more.

Applications developed in CXP Developer are stored in a relational database. To support moving between environments or sharing work, applications can be exported to ML files, an XML-based representation of call flows and dialog steps. This format is completely open and well documented. See the XDK Guide for full documentation.

Bottom line, Voxeo CXP allows you to get your applications into a readable, standardized XML format.

**In other words: no lock-in at the application layer.**

CXP Server, the service execution environment, is a highly scalable, flexible runtime server with features such as:

- Hot redeployment (applying changes to a production service with no downtime)
- Instant restore (backing up to a previously deployed version upon the detection of problems)
- Central cluster management (applying a change to a service with the click of a button, and making changes spread through an entire server cluster)
- Notification framework (sending SNMP traps and/or email notifications on runtime warnings or errors)

Most importantly, CXP Server dynamically renders markup code for phone applications, including VoiceXML, HTML, or custom XML, which is required for SMS, USSD, or IM integration (content markup for these channels is not yet standardized). Through its media platform driver approach, applications developed in CXP are independent of the underlying platform. Much like printer drivers, where documents look the same no matter what printer you use, phone applications will sound or look the same when deployed through CXP on any supported platform. CXP comes equipped with over 150 drivers for VoiceXML + speech recognition environments and continuously adds more drivers for new releases, new markup standards, or entirely new platforms.

Bottom line, CXP is open to any VoiceXML-enabled speech platform; it even supports the deployment of one and the same service on multiple dissimilar platforms in parallel. Of course, CXP is tightly integrated with the Voxeo Prophecy IVR platform and hosting services, which are 100% standards compliant with additional built-in functionality that prevents vendor lock-in.

**In other words: no lock-in at the platform layer.**

CXP Analyzer, the service analysis environment, is based upon Infostore, a database for statistical data that is automatically filled by CXP Server while it processes calls. Its data model is optimized for immediate business intelligence analysis. CXP Analyzer supports leading Business Intelligence (BI) tools such as BusinessObjects, Cognos and MicroStrategy. It comes with over 60 predefined reports that are fully customizable and extendable to suit any reporting needs. The data model of Infostore itself is also extendable, open and well documented. See the Infostore Guide for full documentation.
Bottom line, CXP supports customized logging, while providing a powerful statistical data model out-of-the-box.

**In other words: no lock-in at the reporting and analytics layer.**

For a full overview of the CXP product, please refer to [http://www.voxeo.com](http://www.voxeo.com).

If you are interested in reading our fully public product documentation, please visit [http://help.voxeo.com/go/help](http://help.voxeo.com/go/help).

### 3 SPEEDING MIGRATION TO VOICEXML WITH VOXEO

In addition to being a powerful voice application environment, Voxeo CXP speeds up the task of migrating legacy applications to a VoiceXML-based framework. Note that we say speed up – there is no such thing as a magic button for automatic migration from a proprietary platform to VoiceXML... but if you do it right, you can ease the process while ensuring you don’t find yourself in the same predicament in the future.

#### Development Productivity

First and foremost, CXP Developer simplifies the task of creating phone applications through a variety of features that, by themselves, ease the job of creating callable, automatically documented and deployable applications. CXP is object-oriented. With this comes:

- **Reusability** – Re-use single objects, such as dialog steps, or entire modules, within the same application or across different projects through the use of libraries
- **Inheritance** – Define generic dialog behavior once and have it applied throughout the application. This includes event handling, navigation (i.e. global commands), as well as tuning parameters for ASR, TTS, or general platform tuning.

Generally speaking, CXP applications are modeled and parameterized, not coded. Examples for object types are Output, Input and Menu, where each object represents certain dialog functionality.

- **Output** – Presents output to the caller/user (as audio in the voice channel, video streams in the video channel or as text or images in the text and Web channels)
- **Input** – Comes with an embedded Output object (for re-usability!) to prompt the caller for information, with an embedded Grammar object to activate speech or DTMF grammars, and a mechanism to store user input in a Variable object or to have it directly influence runtime layer conditions
- **Menu** – Comes with an embedded Output object to present an introductory prompt, with menu item outputs to present each item, embedded Grammar objects per item to allow their selection, and a place to provide a target object to tell the Menu object what to do once an item has been selected

In addition to these “standard” objects, other more sophisticated object types such as Confirmation, List, or Pause provide respective dialog functionality to confirm and optionally correct a set of data items collected in the call flow, present a browsable list or even table of information, or put the system on hold during caller think times.

Each object comes with the standard functionality described above plus event handling (reacting to No Input, No Match, Help events, etc.), navigation (allowing global commands such as main menu, help, transfer to agent, etc.), and tuning parameters (to control barge-in, confidence levels, etc.). One object can constitute a fully functional, stand-alone dialog flow by itself.
When comparing the time it takes to create a callable application in plain VoiceXML to what it takes when using CXP, you can be up to 80% more productive. With CXP’s Server-based features such as hot redeployment of services and central cluster management, even more time can be saved when it comes to regular change requests and service updates.

When comparing CXP Developer to other graphical development environments, time is saved using the above-mentioned features. Developer productivity is further increased due to the incorporation of best practices that have been learned working with hundreds of customers over the past 10 years that CXP has been in production.

Remember, once you have developed an application in CXP, it is:

- Automatically documented in PDF and Microsoft Excel (for prompt lists) with the touch of a button
- Deployable on any VoiceXML-enabled speech platform with no further adjustment of the service
- Callable on any phone channel with adjustments only required at the presentation layer; re-use backend connections and business logic, plus the entire creation, testing and debugging environment
- Reportable out-of-the-box through leading Business Intelligence tools or your own reporting suite

We’ve explained how CXP can streamline development and facilitate reusability moving forward. Now let’s explore what you can re-use from your existing applications when moving to CXP.

**CXPML**

The key to using CXP to migrate your existing application lies in CXPML, the XML-based representation format for call flows and dialog steps. It can be used to develop applications or initial building blocks for applications (i.e. the dialog objects), outside of CXP Developer, in a programmatic way. CXPML code can be imported into CXP Developer to ease the developer’s job by serving as the foundation for the call flow.

As an example, consider the following code:
This shows the definition of two objects:

1. **Input object called “Ask for Age”**
   - Contains an initial prompt (<output type="initial">) to say "First, I need to know your age please."
   - Contains a re-prompt (<output type="reprompt">) to say "Please say or type in your age." after an event such as NoInput or NoMatch has occurred
   - Provides a grammar for speech and DTMF recognition to accept numbers (<grammar>)
   - Provides a reference to Variable object “Age”

2. **Variable object called “Age”**
   - Stores the result of what the caller has said or entered
   - Can be used for formatted playback in the dialog

Embedded in a Module that provides generic event handling for NoInput and NoMatch events (which does not need to be repeated in the Input object due to inheritance), a dialog that results from processing this one Input object through CXP server could look like this:

```
“First, I need to know your age please.”

“My age, err, thirty-seven.”

“Excuse me?”

“Thir – ty – se – ven”

“Sorry, again I didn’t get that. Please say or type in your age.”

(types) “3 7”
```

The automatically generated project documentation export for this one object would show its entire configuration in a graphical and coherent way. Note that all properties are derived purely from the small XML snippet above.
In a nutshell: once you have the CXXML representation of your dialog, you have an immediately callable and documented application. And it doesn't take much more to enable this application for SMS, USSD, IM, or the mobile Web.

So how do you arrive at the CXXML code? Basically by applying a script that converts any kind of base material you have in your project to this format. Let’s have a look at different types of code or documentation that you can typically re-use for migration.
Consider the following sample prompt list from an Excel spreadsheet:

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dialog</td>
<td>Name Audiofile</td>
<td>Notes</td>
<td>Reference Visio</td>
<td>Prompt Content</td>
</tr>
<tr>
<td>MEN, Main</td>
<td>MainMenu</td>
<td>[DTMF = Voice]</td>
<td>Page 1, Prompt 2</td>
<td>This is the main menu, please select one of the following options.</td>
</tr>
<tr>
<td></td>
<td>MainMenuReenter</td>
<td>[DTMF = Voice]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CustomerData</td>
<td>CustomerData</td>
<td>[DTMF = Voice]</td>
<td></td>
<td>Customer data.</td>
</tr>
<tr>
<td>MainMenu2ndNi</td>
<td>MainMenu2ndNi</td>
<td>[DTMF = Voice]</td>
<td></td>
<td>I still did not hear you. Please say service plans, customer data, billing, or support.</td>
</tr>
<tr>
<td>MainMenu2ndNm</td>
<td>MainMenu2ndNm</td>
<td>[DTMF = Voice]</td>
<td></td>
<td>I still did not get that. Please say service plans, customer data, billing, or support.</td>
</tr>
<tr>
<td>CC_Type</td>
<td>CreditCardType</td>
<td>[DTMF = Voice]</td>
<td>Page 2, Prompt 1</td>
<td>First, I need your credit card type. Please say Visa, American Express, or Mastercard.</td>
</tr>
<tr>
<td></td>
<td>CreditCardTypeReenter</td>
<td>[DTMF = Voice]</td>
<td></td>
<td>Again I need your credit card type. Please say Visa, American Express, or Mastercard.</td>
</tr>
<tr>
<td></td>
<td>CreditCardType2ndNi</td>
<td>[DTMF = Voice]</td>
<td></td>
<td>I still did not hear you. Please say Visa, American Express, or Mastercard.</td>
</tr>
<tr>
<td></td>
<td>CreditCardType2ndNm</td>
<td>[DTMF = Voice]</td>
<td></td>
<td>I still did not understand. Please say Visa, American Express, or Mastercard.</td>
</tr>
<tr>
<td>CC_Number</td>
<td>CreditCardNumber</td>
<td>[DTMF = Voice]</td>
<td>Page 2, Prompt 2</td>
<td>Please tell me your 16-digit credit card number.</td>
</tr>
<tr>
<td></td>
<td>CreditCardNumberReenter</td>
<td>[DTMF = Voice]</td>
<td></td>
<td>Again I need your 16-digit credit card number, digit by digit.</td>
</tr>
<tr>
<td></td>
<td>CreditCardNumber2ndNi</td>
<td>[DTMF = Voice]</td>
<td></td>
<td>I still did not hear you. Please say or type in your 16-digit credit card number.</td>
</tr>
<tr>
<td></td>
<td>CreditCardNumber2ndNm</td>
<td>[DTMF = Voice]</td>
<td></td>
<td>I still did not understand. Please say or type in your 16-digit credit card number.</td>
</tr>
</tbody>
</table>

Look closer at the documentation above. It includes almost everything you would need for functional dialog objects in a CXP application:

- Audio file names, which can also be used as Audio object names
- Dialog step names, which can be used as Input/Menu/Output object names
- Prompt texts
- Naming conventions (such as “Reenter” or “2ndNi”) that can be exploited to define correct event handling

You might even want to include the notes and Visio references in the description fields of the objects to be created. This way you don’t even lose the documentation references.

All it takes now is writing a VBA script that iterates over all lines, extracts the respective data from the cells, and produces CXPML code. The same approach works with Word-based project documentation. VBA scripts can iterate through all pages and identify the parts where prompt information is coded. If you spend some more effort identifying call flow logic, you can achieve even higher reusability rates.
In the end, the generated CXPML code gets imported into CXP Developer, allowing you to kick-off your implementation with a multitude of pre-configured objects that are ready to be used in your call flow. This is a proven approach that has saved companies with large migration projects hundreds of hours.

**Application Code**

Another source of reusability is the code that represents your existing dialogs. If you currently use a development environment that does not come with any form of readable, text-based (XML or proprietary) code for your applications, then the only viable migration approach for you would be the one just explained, based on documentation. If you do use an IDE that allows an export of applications or already stores call flows in a readable format, the code should contain information that you can use to identify reusable components. By writing a conversion script that takes the code and parses it to find prompt information or any other kind of code that could help you prepare dialog objects, you can produce CXPML. As an example, the Storyboard Manager, an Excel-based prompt management tool integrated into CXP Developer, already comes with such a script to read Nortel PeriPro project files. It is a simple script that extracts all prompt-related information from those files and imports them into a spreadsheet form. As the Storyboard Manager comes equipped with other macros that generate CXPML, you will save a lot of time by not having to manually create the basic building blocks for your application.

While this paper mainly talks about migration to VoiceXML, you might be in a situation in which your applications already are in VoiceXML, but not in an environment like CXP. Through CXPML, you can even leverage plain VoiceXML to get your call flows into the Voxeo CXP platform. CXP also allows you to maintain the main menu or other main modules of your services within its environment, while branching out to existing, plain VoiceXML-based applications at call time if necessary. This helps with step-by-step migration, instead of waiting for the whole set of applications to be converted before you can exploit the great features of CXP.

**Other Resources**

Obviously, getting the call flow migrated is only part of the job. A voice application can include audio files, grammar files, external script code, backend connectors, CTI integration, etc. The good news is that ALL of this can be reused with CXP. Due to its nature of being a runtime application server, it is basically agnostic towards any external resources like the ones mentioned above. It is up to the underlying media platform you use whether you can reuse those files as is, or whether you need to convert them. For conversion of audio files, use any publicly available conversion tools. For conversion of grammar files, use NuGram IDE, an Eclipse plug-in for grammar management bundled with CXP.
Here are some examples of application components and how to reuse them:

**Audio Files**
- Place them on a resource server or just keep them where they are today. Simply configure the URLs in CXP through generic Resource Locator objects.

**Grammar Files**
- Place them on a resource server or keep them where they are today. Configure the URLs in CXP through generic Resource Locator objects.

**Script Code**
- Place them on a resource server or keep them where they are today. Configure the URLs in CXP through generic Resource Locator objects.
- CXP ships with an ECMAScript engine. If this is not the programming language you are used to or using in your current environment, CXP offers a flexible Script Bus architecture to allow plugging in any script engine you need.

**Backend connectors**
- CXP provides four types of connectors: Java, CGI/HTTP, Web Services, and database. If you already use Java or server-based CGI scripts/Web services to access backend systems, you are good to go. If not, add a Java wrapper around it. If you have a relational database system in use, simply communicate with it over a JDBC connection.

**Migration Tools and Partners**

For those migrating from certain platforms, Voxeo has conversion tools and partnerships in place that make the process even more streamlined. CXP Migration Manager can convert about 95% of Nortel PeriPro building blocks over to Voxeo CXP, reducing the hand-coding needed to complete the conversion to an absolute minimum. If you are moving from Edify or Intervoice IQTalk, Voxeo has a certified partner whose conversion tool, Arca+, typically results in a 50% cost reduction and 75% time savings compared to rewriting your applications.

**Migration Lessons Learned**

We want to be honest. There is no magic bullet for migration away from your current infrastructure. But if you are facing platform end-of-life or working with a sub-par VoiceXML platform, the investment will pay off. In this whitepaper we summarize what we believe is a worthwhile effort even if you do not immediately face end-of-life of your current platform – and why it is a good idea to choose Voxeo.

Using CXP will result in more stable, feature-rich applications and save you time in the mid- to long-term with automatic documentation, automatic reporting, centralized cluster deployment, and many more features. CXpML is an open, flexible format for call flow representation that is ready to be exploited through automatic conversion scripts to help you simplify migration and save money.
Now that we’ve discussed the CXP and application migration, let’s back up because choosing the right service creation environment is only one piece of the overall solution. You still need a platform for your application. It’s critical to recognize that many platform providers claim openness, but use proprietary development tools and tags to maintain ownership of your application. As mentioned previously, Voxeo CXP is a platform-independent solution that enables one-click deployment to virtually every modern IVR platform offered today. That said, Voxeo’s own Prophecy platform and Prophecy Hosting services offer compelling advantages over other IVR platforms:

**Standards Compliance:** Voxeo’s architecture is based entirely on open standards such as VoiceXML 2.0 and 2.1, CCXML, SIP, MRCP and SSML.

**You own your application:** With Voxeo, you own your intellectual property, resulting in the flexibility to switch providers, bring your application in house, or engage your choice of development resources.

**Compatibility:** In addition to the functionality offered by CXP, the Voxeo Prophecy platform itself includes compatibility modes that make porting VoiceXML IVR applications as easy as changing a single Prophecy setting.

**Third-party development tools:** Voxeo Prophecy and Prophecy Hosting give you the flexibility to use CXP or any VoiceXML compliant third-party development tool.

**Multi-channel:** In the same way that CXP supports “develop once, deploy anywhere” functionality for multi-channel phone applications, the Voxeo platform meets your Unified Self-Service needs with existing support for voice, SMS, Twitter and mobile web communications.

**Speech flexibility:** The Voxeo platform includes built-in speech recognition and speech synthesis (ASR and TTS) engines, and supports any MRCP-compliant third-party speech technology.

**Deployment flexibility:** Unlike other vendors, Voxeo offers both on-demand and on-premise deployment options with the ability to easily move from one to the other or leverage a hybrid model. For instance, some customers deploy a solution on-premise and use Voxeo hosting for overflow and disaster recovery. Voxeo’s on-premise Prophecy platform is the very same product used to power Voxeo Prophecy Hosting.

**Telephony options:** Voxeo’s IVR platform is natively SIP-based. Customers can connect to the platform either directly through VoIP/SIP or via traditional TDM circuits via VoIP gateways. Voxeo owns its SIP stack and can ensure compatibility with any SIP-based environment as needed.

**Community:** With a standards-based platform, finding qualified developers for new applications is easier and more cost-efficient. Additionally, the Voxeo Connect community of certified partners gives you access to upgrades like voice biometrics, professional voice prompts, and custom-built solutions that align with evolving customer contact preferences, and integrate seamlessly with your platform.
Get in touch with us today to find out how you can leverage your existing source code and documentation for faster and easier VoiceXML migration. Better yet, find out how you can unlock the potential of your communications applications with a flexible, robust, and standards-based solution. At Voxeo, we choose to keep our customers by providing a better product and better support – never by locking them in. Contact us today!

To learn more, visit www.voxeo.com for next steps and links to get started.

Visit us or join our conversations on the web at
www.voxeo.com, blogs.voxeo.com, or twitter.com/voxeo

Contact us at:

**VOXEEO CORPORATION**
USA: +1 407 418 1800
Europe/UK: +44 20 7887 6085

Germany: +49 2203 1857 300
Asia: +86 10 8282 5011
sales@voxeo.com

© 2012 Voxeo Corporation