





XDK Guide

VoiceObjects 9.1

To ensure that you are using the documentation that corresponds to the VoiceObjects software you are licensed to use, compare this version number with the software version shown in the Help menu of the VoiceObjects software you are using.

Copyright

Copyright © 2001-2010 Voxeo Germany GmbH and its licensors. All rights reserved.

Published in Germany – Legal information 2009

Information in this document is subject to change without notice and does not represent a commitment on the part of Voxeo or any of its affiliated companies (collectively “Voxeo”). The software described in this document is furnished solely under a license agreement. The software may be used or copied only in accordance with the terms of the license agreement. You shall not reverse engineer the software or sub-license, distribute or make the software available to third parties by other means except as specifically allowed in the license agreement or by mandatory law. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording, for any purpose, without the express written permission of Voxeo.

Protected by German, European and US patents. Further patents pending.

Companies, names, and dates used in examples herein are fictitious unless otherwise noted. If such names affect copyrights or trademarks or others, please notify Voxeo by e-mail at support@voxeo.com.

Trademarks

VoiceObjects is a registered trademark. Any other trademarks, trade names or service marks mentioned in this document belong to their respective owners.

The material presented herein is based upon information that we consider reliable, but we do not represent that it is error-free and complete. Voxeo is not making any representation or granting any warranty with respect to such material, and the distribution of such material shall not subject Voxeo to any liability.

Third-Party Products

If Licensed Products are distributed together with third-party software or if this is contained in the Licensed Products, which are subject to additional license provisions, Licensee undertakes to observe the license provisions of the third party.

Explicit Copyright Notice

The VoiceObjects software includes software developed by the Apache Software Foundation (www.apache.org). Copyright © 1999-2010 – The Apache Software Foundation. All rights reserved.

Java and all Java-related trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S., other countries, or both.

Specific versions of the VoiceObjects software contain copyright material authorized by the Eclipse Foundation (www.eclipse.org), their contributors and others. All rights reserved.

Specific versions of the VoiceObjects software work with Microsoft Excel or make use of copyright material from Microsoft Corporation (www.microsoft.com). All rights to such copyright material rest with Microsoft. Microsoft and Excel are registered trademarks of Microsoft Corporation.

The VoiceObjects software is bundled with the software *NuGram IDE Basic Edition* developed by Nu Echo Inc. (www.nuecho.com). All title and copyrights in and to the software *NuGram IDE Basic Edition*, the accompanying printed materials, and any copies of the software are owned by Nu Echo Inc. and its suppliers.

Document Number: E-013-20100208-VO9



Table of Contents

TABLE OF CONTENTS	3
XDK OVERVIEW	4
Introduction	4
Application Development.....	4
Top-level elements	5
Embedded elements.....	7
Standard attributes	7
Object references	9
Application Deployment.....	9
VoiceObjects Desktop deployment	9
Web services deployment	10
Configuration Objects	10
Servers	10
Services	11
Users and access control	16
Projects.....	18
Importing into the VoiceObjects Metadata Repository	19



XDK Overview

Introduction

With XDK, the XML Development Kit, VoiceObjects provides the voice market's first open, XML-based interface to its phone application server. XDK enables developers to use the Integrated Development Environment (IDE) of their choice, such as Eclipse, SAP NetWeaver Studio, or IBM WebSphere Studio, while still retaining full access to the object-oriented design approach provided by VoiceObjects.

At the heart of XDK stands VoiceObjectsXML, an XML-based application-level markup language. It sits on top of open standard such as VoiceXML or SALT, yet is independent of individual media platforms, owing to the VoiceObjects media platform driver concept. Furthermore, it provides all the powerful capabilities built into VoiceObjects' patented object technology such as a true object-oriented development approach and inheritance.

Application Development

Applications may be developed in XDK using any Integrated Development Environment (IDE) of choice. A Document Type Description (DTD) and XML schema are provided to enable validation and such features as auto-completion.

A VoiceObjectsXML application definition may consist of multiple files, one of which needs to be the root file from which all others are referenced. Additional files can be imported using the following notation:

```
<xi:include href="../../../lib/myobjects.xml"/>
```

The `href` reference may use relative or absolute file or HTTP URLs. Relative URLs are resolved relative to the location of the parent file. Imports may only be defined at the beginning of a file, before any object definition. Objects may be referenced across different files. Before performing a validity check, all files are merged to determine the total set of objects.

Each individual document needs to be wrapped in

```
<VoiceObjectsXML version="9.1"> ... </VoiceObjectsXML>
```



Caution: All referenced VoiceObjectsXML files must have the same version information set. A mix of different VoiceObjects versions is not supported as it might lead to problems on importing and deploying.

The file encoding needs to be specified in the `<xml>` header line, e.g.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
```



Top-level elements

The following table lists the available top-level VoiceObjectsXML elements. They correspond to the objects of the same name, and are described in detail in the *Object Reference*.

Element	Description
<audio>	Defines an Audio object. Uses embedded <altText>.
<businessTask>	Defines a Business Task object.
<case>	Defines a Case object. Uses embedded <caseItem>.
<collection>	Defines a Collection object.
<confirmation>	Defines a Confirmation object. Uses embedded <correctionItem>.
<connector>	Defines a Connector object. Uses embedded <connectorItem>.
<database>	Defines a Database object.
<exit>	Defines an Exit object.
<expression>	Defines an Expression object. Uses embedded <argument>.
<format>	Defines a Format object. Uses embedded <formatItem>.
<goto>	Defines a Goto object.
<grammar>	Defines a Grammar object. Uses embedded <grammarItem>.
<hyperlink>	Defines a Hyperlink object.
<if>	Defines an If object.
<input>	Defines an Input object. Uses embedded <resultHandling>.
<layer>	Defines a Layer object. Uses embedded <layerState>.



Element	Description
<list>	Defines a List object. Uses embedded <contentFormatting>, <columnGroup>, <navigationMessages>, <selectionCommands>.
<log>	Defines a Log object. Uses embedded <logItem>.
<loop>	Defines a Loop object.
<menu>	Defines a Menu object. Uses embedded <menuItem>.
<module>	Defines a Module object.
<osdm>	Defines an OSDM object. Uses embedded <osdmItems>.
<output>	Defines an Output object. Uses embedded <outputItem>.
<pause>	Defines a Pause object.
<plugin>	Defines a Plug-In object. Uses embedded <parameterSet>, <value>.
<recording>	Defines a Recording object. Uses embedded <parameterSet>.
<resourceLocator>	Defines a Resource Locator object.
<script>	Defines a Script object. Uses embedded <parameterSet>, <value>.
<sequence>	Defines a Sequence object.
<silence>	Defines a Silence object.
<text>	Defines a Text object.
<transfer>	Defines a Transfer object. Uses embedded <parameterSet>.
<variable>	Defines a Variable object.
<video>	Defines a Video object.



Embedded elements

The top-level elements make use of the following embedded elements:

Element	Description
<argument>	Defines arguments for expressions.
<eventHandling>	Defines event handling.
<item>	Defines individual entries in parameter sets or result handling.
<osdmItems>	Defines parameterization for OSDM objects.
<parameterSet>	Defines parameter sets.
<resultHandling>	Defines result handling for inputs.
<standardNavigation>	Defines standard navigation.
<tuning>	Defines tuning settings.
<value>	Wraps embedded code for plug-ins or scripts, or embedded values for variables, collections, and expression arguments.

Standard attributes

The top-level VoiceObjectsXML elements carry certain standard attributes, all of which are optional. The following table provides an overview of these attributes and their uses.



Attributes listed in the descriptions for the individual XML elements are to be seen in addition to the standard attributes. They are optional unless marked as "required".

Attribute	Description
name	Name of the object. Can be used to refer to the object within XDK files. No two elements of the same type may have the same name.
referenceID	Reference ID of the object. A text string without blanks or special characters other than underscores. Maximum length is 256 bytes. No two elements in an application may have the same reference ID, regardless of type.
guid	Globally unique ID for the object. Unless this is specified, a new ID is generated whenever a VoiceObjectsXML application is deployed to VoiceObjects Server or imported into the VoiceObjects Metadata Repository. GUIDs are conceptually like reference IDs, and an



Attribute	Description
	object may have the same reference ID and GUID. No two elements in an application may have the same GUID, regardless of type. GUIDs <u>must</u> be of the form OVAPxxxxx where xxxxx is a string of exactly 40 characters from among a-z, A-Z, 0-9.
shortDescription	A text string defining the short description of the object. Maximum length is 4,000 bytes.
comment	A text string defining a comment for the object. Maximum length is 8,000 bytes.
keywords	A text string defining the keywords for the object (for searching). Maximum length is 4,000 bytes.
versionDescription	A text string defining the version description of the object. Maximum length is 8,000 bytes.
errorDescription	A text string defining the error description of the object. Maximum length is 4,000 bytes.
disable	Either <i>true</i> or <i>false</i> . Indicates whether the object is disabled or not. If not specified, defaults to <i>false</i> . This attribute is not available for those object types that cannot be disabled (<format>, <resourceLocator>, <variable>, <collection>, <layer>).
link	A reference to another object of the same type. References begin with the # character and may use either the object's name or the object's reference ID. Objects may be referenced across file boundaries. When resolving references, an object of the same type with the given string as name is searched for first. If no such object exists, then an object of the same type with the given string as reference ID is searched for. If no such object exists either, an error message is produced. If a link is specified, all other attributes (except <i>usage</i> and <i>type</i>), and all children are ignored. Objects from libraries can be linked using the notation <i>object://LibraryName/LibraryVersionName#ObjectName</i>
usage	Indicates how this element is used as a child of another element. This is relevant when an element may have several children of the same type. Legal usage types are listed in the DTD and indicated in this document. This attribute is only available on those object types that require it.



-  **Note:** In case of the attributes `name`, `shortDescription`, `comment`, `keywords`, `versionDescription`, and `errorDescription` using UTF-8 characters is allowed, but may reduce the given size limit, as some characters use more than one byte.
-  **Note:** When using MaxDB as the Metadata Repository DBMS the following limits apply to the size of the standard properties described above:
`referenceID: 255 bytes; shortDescription: 500 bytes; comment: 500 bytes; keywords: 255 bytes; versionDescription: 500 bytes; errorDescription: 255 bytes.`

Object references

Objects are referenced in VoiceObjectsXML by their name to enhance readability. Since most references combine object type and object name as e.g. in

```
<sequence>
  <output link="#Welcome message"/>
  <input link="#Enter account number"/>
</sequence>
```

there is no confusion because no two objects of the same type may have the same name.

When objects are referenced in attributes, a "type hint" can optionally be added to eliminate confusion between objects of equal name but different type:

```
<list selectionCart="/**collection**/#Cart collection">
```

Objects from libraries can be referenced using the fully qualified object notation:

```
<module name="Prime Banking">
  <output link="object://General library/Version1#Time-
dependent greeting"/>
  ...
</module>
```

Application Deployment

Applications developed within XDK are deployed to VoiceObjects Server as services. This can be done using VoiceObjects Desktop and the Web Services Interface.



Caution: When using Infostore in conjunction with XDK applications, make sure to provide reference IDs and names for all Module objects in your application. Otherwise, reports on application usage will suffer from inconsistent object IDs.

VoiceObjects Desktop deployment

XDK applications can be referenced directly from within services created within VoiceObjects Desktop by using a URI as start object definition. Upon (re-) deployment of the service, the application is dynamically loaded by the server.



For more information, refer to the *Deployment Guide*.

Web services deployment

Services can be deployed through the Web Services Interface as standard services using the `deployXDKService` method, or as volatile services using the `redeployXDKApplication` method. The difference in the latter case is that no intermediate files need to be generated from which the application definition is loaded.

For more details, refer to the *Web Services Guide*.

Configuration Objects

All VoiceObjects configuration objects (servers, services, users, and projects) can also be defined using VoiceObjectsXML markup. The subsequent paragraphs describe how to define the different object types.



Note: Only a single type of configuration object may be defined within one XDK file, but multiple objects of the same type can be defined at once.

Servers

Servers are configured using the `<server>` element. It has two children `<serviceList>` (described below) and `<accessControlList>` (described in Users and access control) and the following attributes (in addition to the standard attributes, of which `referenceID` is mandatory):

Attribute	Description
<code>sessionLimit</code>	Defines the maximum number of concurrent sessions for this server.
<code>sessionGuarantee</code>	Defines the guaranteed number of concurrent sessions for this server.
<code>systemDBLogging</code>	Defines whether System DB logging is enabled for this server. Either <i>true</i> or <i>false</i> . If not specified, defaults to <i>true</i> .
<code>customDBLogging</code>	Defines whether Custom DB logging is enabled for this server. Either <i>true</i> or <i>false</i> . If not specified, defaults to <i>true</i> .
<code>utteranceRecording</code>	Defines whether utterance recording is enabled for this server. Either <i>true</i> or <i>false</i> . If not specified, defaults to <i>true</i> .

`<serviceList>` may contain any number of `<service>` children defining the list of services hosted on the server. The services can either be defined inline, or they can be referenced using the `link` attribute. When a service is referenced using the `link` attribute, the corresponding Service object must already exist in the Metadata Repository.



- i** » **Note:** When using an inline definition of a service, this service must not exist in the Metadata Repository. Otherwise an error message will be returned when importing the server definition.

Any number of servers may be defined in a single XDK file.

Example:

```
<server name="Production Server" referenceID="VOserver"
  systemDBLogging="true" customDBLogging="false"/>
```

Services

Services are configured using the `<service>` element. It has one child `<accessControlList>` (described in Users and access control) and the following attributes:

Attribute	Description
name	Name of the Service object (for display in VoiceObjects Desktop).
vsn	VoiceObjects Service Name for the service. Mandatory.
link	A reference to another service, either by name or by VSN. References begin with the # character. If a link is specified, all other attributes (except <i>usage</i> and <i>type</i>), and all children are ignored.
startObject	Start object for the service, to be provided as a URL. Note that the start object itself must be designated by its reference ID, not by its name.
enableValidation	Indicates whether the application should be validated when loading the cache. Either <i>true</i> or <i>false</i> . If not specified, defaults to <i>true</i> .
retainOnShutdown	Indicates whether the application cache should be persisted even when the server is completely shut down. Either <i>true</i> or <i>false</i> . If not specified, defaults to <i>false</i> .
cacheStandby	Indicates whether the application cache for this service may be dropped from memory. Either <i>onDemand</i> or <i>never</i> . If not specified, defaults to <i>onDemand</i> .
dnisMapping	Comma-separated list of DNIS that map to this service.



Attribute	Description
startupMode	Defines the start-up mode for the service. One of <i>automatic</i> , <i>disabled</i> , <i>manual</i> . If not specified, defaults to <i>automatic</i> .
respectUserAgent	Indicates whether the server should attempt to derive the appropriate driver from the user-agent information contained in the initial request. Either <i>true</i> or <i>false</i> . If not specified, defaults to <i>false</i> .
language	Defines the initial language setting for the service. Can be <i>default</i> or a valid language code (e.g. <i>de-DE</i> , <i>en-US</i> , etc.). If not specified, defaults to <i>default</i> . Appendix A – Language Codes in the <i>Object Reference</i> contains a list of all language codes available in VoiceObjects together with the respective language they represent.
driver	Defines the media platform driver for the service. Can be <i>default</i> or a valid media platform driver code, e.g. <i>Avaya_VP_4_0_VXML_2_1_ASR_Nuance</i> (see the <i>Deployment Guide</i> or the DTD or XSD for a list of all available drivers). If not specified, defaults to <i>default</i> .
sessionLimit	Defines the maximum number of concurrent dialog sessions that is allowed for this service.
sessionGuarantee	Defines the number of guaranteed sessions that should be set aside for this service.
sessionDistribution	Defines the distribution of dialog sessions between services within the same custom site. Either <i>disabled</i> , or a float value between 0 and 1 indicating a percentage (e.g. 0.75 for 75%). If not specified, defaults to <i>disabled</i> .
maintainer	E-mail address of the application administrator. Is used by the media platform to send e-mail in case of errors.
loggingDestination	Defines the default logging destination for Custom DB logging. One of <i>file</i> , <i>customDB</i> , <i>mediaPlatform</i> , <i>systemDB</i> , <i>logOSDM</i> . If not specified, defaults to <i>file</i> .
loggingCategories	Defines the active logging categories for Custom DB logging. Comma-separated list of values from among <i>info</i> , <i>debug</i> , <i>error</i> , <i>statistics</i> , <i>billing</i> . If not specified, all of the categories are enabled.



Attribute	Description
systemDBLogging	Defines whether System DB logging is enabled. Either <i>true</i> or <i>false</i> . If not specified, defaults to <i>false</i> .
customDBLogging	Defines whether Custom DB logging is enabled. Either <i>true</i> or <i>false</i> . If not specified, defaults to <i>false</i> .
inputStateLogging	Defines whether Input State logging is enabled. Either <i>true</i> or <i>false</i> . If not specified, defaults to <i>false</i> .
loggingCoverage	Defines the percentage of dialogs that should be written to System DB logging. Can be a float value between 0 and 1 indicating a percentage (e.g. 0.75 for 75%). If not specified, defaults to 1.
filterScope	Defines whether the logging coverage extends only to input states, to System DB logging, or to System and Custom DB logging. One of <i>inputState</i> , <i>systemDB</i> or <i>systemDBCustomDB</i> . If not specified, defaults to <i>systemDB</i> .
enableUtteranceRecording	Defines whether utterance recording is enabled. Either <i>true</i> or <i>false</i> . If not specified, defaults to <i>false</i> .
recordUtterances	Defines the default value for the <code>recordUtterances</code> setting on dialog objects. Either <i>true</i> or <i>false</i> . If not specified, defaults to <i>true</i> .
recordingScope	Defines whether utterance recording files are created only for recognized utterances (<i>recognition</i>), for utterances that lead to No Match events (<i>noMatch</i>), or for both (<i>all</i>). If not specified, defaults to <i>all</i> .
utteranceRecordingPath	Defines the path that utterance recording files are written to.
utteranceRecordingURL	Defines the URL through which utterance recording files can be retrieved.
recordingCoverage	Defines the percentage of dialogs that should write utterance recording files. Can be a float value between 0 and 1 indicating a percentage (e.g. 0.75 for 75%). If not specified, defaults to 1.
bargein	Defines the default value for barge-in behavior. Either <i>true</i> or <i>false</i> . If not specified, defaults to <i>true</i> .



Attribute	Description
<code>grammarType</code>	Defines the default grammar type. Must be one of <i>default</i> , <i>abnf</i> , <i>gsl</i> , <i>jsgf</i> , <i>xml</i> , <i>cisco</i> , <i>precompiled</i> , <i>none</i> . If not specified, defaults to <i>default</i> .
<code>grammarPrecedence</code>	Defines the precedence between embedded and external grammar definitions within Grammar objects. Either <i>embedded:external</i> or <i>external:embedded</i> . If not specified, defaults to <i>embedded:external</i> .
<code>outputMode</code>	Defines the default output mode. Either <i>audio:tts</i> or <i>tts:audio</i> . If not specified, defaults to <i>audio:tts</i> .
<code>inputMode</code>	Defines the default input mode. One of <i>voicedtmf</i> , <i>voice</i> , or <i>dtmf</i> . If not specified, defaults to <i>voicedtmf</i> .
<code>exitType</code>	Defines the default dialog exit type. Can be <i>disconnect</i> , <i>exit</i> , or <i>return</i> . If not specified, defaults to <i>disconnect</i> .
<code>videoType</code>	Defines the default video type. One of <i>default</i> , <i>video</i> , <i>videoaudio</i> , <i>none</i> . If not specified, defaults to <i>default</i> .
<code>dialogTimeout</code>	Defines the dialog timeout. An integer value that is interpreted as seconds. If not specified, defaults to <i>300</i> (5 minutes).
<code>standbyTimeout</code>	Defines the default standby timeout. An integer value that is interpreted as seconds. If not specified, defaults to <i>600</i> (10 minutes).
<code>connectorTimeout</code>	Defines the default connector timeout. An integer value that is interpreted as seconds. If not specified, defaults to <i>60</i> (1 minute).
<code>standardNavigationHistory</code>	Defines the length of the dialog navigation history for the service. Can be <i>disabled</i> , or an integer between 1 and 10. If not specified, defaults to 3.
<code>audioFileExtension</code>	Defines the standard audio file extension. Legal values are <i>none</i> , <i>wav</i> , <i>aif</i> , <i>aiff</i> , <i>dwd</i> , <i>mp3</i> , <i>snd</i> , <i>au</i> , <i>voc</i> , <i>vox</i> . If not specified, defaults to <i>none</i> .
<code>grammarFileExtension</code>	Defines the standard grammar file extension. Legal values are <i>none</i> , <i>abnf</i> , <i>grammar</i> , <i>grm</i> , <i>grxml</i> , <i>gsl</i> , <i>jsgf</i> . If not specified, defaults to <i>none</i> .



Attribute	Description
<code>videoFileExtension</code>	Defines the standard video file extension. Legal values are <i>none</i> , <i>3gp</i> , <i>3g2</i> , <i>mp4</i> , <i>mp4v</i> , <i>m4v</i> , <i>rtp</i> , <i>bmp</i> , <i>gif</i> , <i>jpg</i> , <i>jpeg</i> . If not specified, defaults to <i>none</i> .
<code>textFileExtension</code>	Defines the standard text file extension. Legal values are <i>none</i> , <i>txt</i> . If not specified, defaults to <i>none</i> .
<code>rejectResponse</code>	Defines the response behavior for dialogs that are rejected due to missing licenses. Either <i>overflow</i> or <i>redirect</i> . If not specified, defaults to <i>overflow</i> .
<code>proxyHost</code>	Defines a proxy host to use for HTTP connectors. If not specified, no proxy is used.
<code>proxyPort</code>	Defines a proxy port to use in combination with the proxy host. Only relevant if <code>proxyHost</code> is defined.
<code>resourceLocatorPath</code>	Defines a physical path that overwrites the respective entries in resource locators with the option <i>allowServiceOverwrite</i> enabled. If not specified, the local entries of all resource locators are used.
<code>loopAbortHandlerURL</code>	Defines the location of a file holding valid VoiceXML code that will be used as the loop abort handler.
<code>configurationURL</code>	Defines the location of a configuration file that defines application default values.
<code>resourceLocatorURL</code>	Defines a URL that overwrites the respective entries in resource locators with the option <i>allowServiceOverwrite</i> enabled. If not specified, the local entries of all resource locators are used.
<code>osdmURL</code>	Defines a URL that points to the OSDM installation to be used by OSDM objects.
<code>osdmEventLogging</code>	Indicates whether OSDM event logging is enabled. Either <i>true</i> or <i>false</i> . If not specified, defaults to <i>false</i> .
<code>osdmEventLogDestination</code>	Defines the file to which OSDM event logging should be written. If not defined, the default OSDM event logging destination as defined by the OSDM installation is used.



Attribute	Description
shortDescription	A text string defining the short description of the service.
errorDescription	A text string defining the error description of the service. Maximum length is 4,000 bytes.
versionDescription	A text string defining the version description of the service. Maximum length is 8,000 bytes.
comment	A text string defining the comment for the service.
keywords	A text string defining the keywords for the service.

Multiple services may be defined in a single XDK file, wrapped in the `<serviceList>` element.

Example:

```
<service vsn="insurance"  
startObject="http://myserver/XDK/PrimeInsurancePortal.xml#Portal"  
language="en-US" systemDBLogging="true"/>
```

Users and access control

Users are configured using the `<user>` element. It has one child `<accessControlList>` (described below) and the following attributes (in addition to the standard attributes):

Attribute	Description
userID	Defines the user ID for this user. This replaces the standard <code>referenceID</code> attribute, and is required.
password	Defines the password for this user.
email	Defines the e-mail address for this user
role	Defines the user role. One of <i>reviewer</i> , <i>designer</i> , <i>serviceController</i> , <i>observer</i> , <i>siteController</i> , <i>siteAdministrator</i> , <i>userManager</i> , <i>serverController</i> , <i>serverAdministrator</i> . If not specified, defaults to <i>designer</i> .
accountStatus	Defines whether the account is active or not. Either <i>active</i> or <i>inactive</i> . If not specified, defaults to <i>active</i> .



Attribute	Description
allowPasswordChange	Defines whether the user is allowed to change his password (<i>true</i> or <i>false</i>), or whether the password needs to be changed on next login (<i>login</i>). If not specified, defaults to <i>login</i> .
allowProjectCreation	Defines whether the user is allowed to create new projects. Either <i>true</i> or <i>false</i> . Note that this value may be irrelevant, depending on the selected user role. If not specified, defaults to <i>true</i> .
allowWSIAccess	Defines whether the user is allowed to access the Web Services Interface. Either <i>true</i> or <i>false</i> . If not specified, defaults to <i>true</i> .
auditDesktop	Defines the audit level for user interactions with VoiceObjects Desktop. One of <i>none</i> , <i>all</i> , <i>denied</i> . If not specified, defaults to <i>none</i> .
auditServer	Defines the audit level for user interactions with VoiceObjects Server. One of <i>none</i> , <i>all</i> , <i>denied</i> . If not specified, defaults to <i>denied</i> .
auditWSI	Defines the audit level for user interactions with the Web Services Interface. One of <i>none</i> , <i>all</i> , <i>denied</i> . If not specified, defaults to <i>denied</i> .
The attributes listed below are only relevant if the user is the root Site Administrator for a custom site. In this case, they define settings that are applied in the entire site.	
siteID	Defines the name of the site. If not defined, the user ID of the root Site Administrator of the site is used.
basePath	Defines a base physical path used for all resource locators in applications within the site.
baseURL	Defines a base URL used for all Audio, Video, Grammar, and Plug-In objects in applications within the site.
connectorURL	Defines a base URL used for all Connector and Script objects in applications within the site.
importPath	Defines a base import/export path used for the site.
startObjectPath	Defines a base URL used for the start objects of all XDK-based services in the site.
utteranceRecordingPath	Defines the base path used for writing utterance recording files in the site.



Attribute	Description
utteranceRecordingURL	Defines the base URL through which utterance recording files can be retrieved in the site.
sessionGuarantee	Defines the guaranteed number of concurrent sessions for the site.
sessionLimit	Defines the maximum number of concurrent sessions for the site.
serverLimit	Defines the maximum number of concurrently active servers for the site.
serviceLimit	Defines the maximum number of concurrently active services for the site.
designerSeats	Defines the maximum number of Designer seats available within the site.
siteSeats	Defines the maximum number of Site seats available within the site.
seats	Defines the maximum number of VoiceObjects Desktop seats available within the site.
osdm	Indicates whether OSDM support is available within the site. Either <i>true</i> or <i>false</i> . If not specified, defaults to <i>false</i> .
infostore	Indicates whether Infostore is available within the site. Either <i>true</i> or <i>false</i> . If not specified, defaults to <i>true</i> .

Access control lists for objects are defined using the `<accessControlList>` element. It has one child `<aclEntry>`, which in turn has an attribute `user` that references a User object through its `userID`.

Any number of users may be defined in a single XDK file.

Example:

```
<user name="John Smith" userID="jsmith" password="uhg6df8"
  role="serviceController"/>
<user name="Site Alpha" userID="adminAlpha" password="alpha"
  siteID="SiteAlpha" sessionGuarantee="5" sessionLimit="20"/>

<accessControlList><aclEntry user="jsmith"/></accessControlList>
```

Projects

Projects are configured using the `<project>` element. It has two children `<accessControlList>` (optional) and `<projectVersion>` (mandatory), and the standard attributes of which `name` is mandatory.

In addition, the `<project>` element has an attribute `isLibrary`, which can be *true*



or *false* (the default). The `isLibrary` attribute is only interpreted the first time a project is created and cannot be modified later on.

A project version is configured using the `<projectVersion>` element. It has one optional child `<libraries>` and the standard attributes, of which `name` is mandatory. When creating a project, exactly one project version must be created. New project versions can only be created by publishing existing ones. For more information, refer to the *Web Services Guide*.

Libraries are attached to project versions using the `<libraries>` element. It may contain one or more `<library>` children. Each `<library>` has two mandatory attributes `libraryName` and `libraryVersionName`.

Any number of projects may be defined in a single XDK file.

Example:

```
<project name="Hello World" shortDescription="My first application">
  <projectVersion name="1.0"/>
</project>
<project name="Credit Card Validation">
  <accessControlList>
    <aclEntry user="john17"/>
    <aclEntry user="alice32"/>
    <aclEntry user="bob19"/>
  </accessControlList>
  <projectVersion name="1.0">
    <libraries>
      <library libraryName="Common tasks" libraryVersionName="3.6"/>
    </libraries>
  </projectVersion>
</project>
```

Importing into the VoiceObjects Metadata Repository

Applications developed in XDK can be imported into the VoiceObjects Metadata Repository through VoiceObjects Desktop or through the Web Services Interface. This is done using the standard import mechanism described in Chapter 7 – *Basic Commands* in the *Desktop for Eclipse Guide* and *Desktop for Web Guide*, and in the *Web Services Guide*.



Tip: When you intend to import your XDK application into the VoiceObjects Metadata Repository, you should define the *name* attribute for all objects. For objects that do not have a name defined, a name is constructed during the import process based on the name of the XDK file and the line number in which the object definition starts (e.g. HelloWorld.xml_17). The resulting application is fully functional, but not very readable.

In addition to applications, it is also possible to import configuration objects into the Metadata Repository.