

# VOXI Command API

**Confidential**



Introduction.....	4
Databases.....	4
Simultaneous access.....	4
Security/Authentication.....	4
Configuring VOXI.....	5
Installation.....	5
Database.....	5
Port.....	5
Command lockouts.....	5
Maximum active processing engines.....	5
Other VOXI settings.....	5
Trace purging.....	5
Engine retirement.....	6
Overhead logging.....	6
Command logging.....	6
Other.....	6
Monitoring.....	6
VOXI GET Requests.....	8
VOXI Requests and Results.....	9
Requests.....	9
Results.....	10
Metadata.....	11
General.....	11
System.....	13
Options.....	14
Authentication (user).....	15
Messages.....	18
Sending.....	18
Retrieving/viewing.....	18
News.....	21
Calendar.....	24
Volunteer profile.....	27
Account Setup.....	35
Applications.....	36
Applications – Custom.....	40
Job definitions.....	45
Job group definitions.....	50
Job signup.....	52
Slots.....	57
Locations.....	61
Lists.....	66
Assignment checkin/checkout.....	68
Checkin/Hours entry.....	71
Hours.....	73
Hours Pending.....	76
Hours Types.....	79
Hours Charge Codes.....	81
Assignments.....	83
Training.....	87
Surveys.....	89
Community Partners.....	91
Community Partner Entrants.....	95
Community Partner Events.....	97
Community Partner Owners.....	100
Corporate Teams.....	101
Kiosk Locations.....	104
Customer management.....	107
Localization.....	111

Knowledgebase and Support..... 112  
Logs ..... 114  
VOXI Helper..... 115  
    Configuring VOXI Helper ..... 115  
    VOXI Helper as a Service..... 115

## Introduction

VOXI, the VSys One eXchange Interface, provides an API to the business logic and data stored in VSys One and GMS 6. All communication with VOXI is done using HTTP on a port specified in the `VSys.ini` file.

VOXI is a multithreaded Delphi application built with the same code as VSys One and GMS 6. This allows changes and enhancements made to these applications to be trivially available to VOXI and the tools which call it.

## Databases

External applications do not need, or in general want, to directly access the data tables use by VSys One, GMS 6 and VOXI. Unless otherwise indicated, the data stored in these tables is not documented nor intended for direct access. Where it is documented, it may be used for querying but not updating or deleting.

## Simultaneous access

By default, five simultaneous requests can be processed by VOXI; any additional requests are queued while the others are processed (this setting is not configurable at this time). If VOXI is unable to process an incoming request within ten seconds it will respond with a

```
503: SERVER TOO DAMNED BUSY TO ALLOCATE A PROCESSING ENGINE FOR YOU (xx IN USE)
```

where “xx” is the number of active processors in use.

Where feasible, if an application or tool has multiple requests or actions to be performed, submit these in bulk in a single VOXI request. This uses only one command processor and the requests are handled sequentially, greatly reducing the load on VOXI.

VOXI is essentially stateless: all information that it carries is posted to the database immediately, and little information is cached between requests. This means that you can have a farm of VOXI instances running on different machines, all essentially identical and returning the same data. That said, VOXI does not provide implicit locking. Editing a person, posting hours, registering for courses: it makes no attempt to prevent multiple instances or even multiple sessions on the same VOXI instance from doing something rude and self-destructive by making simultaneous changes to the same person.

## Security/Authentication

Access to VOXI is unauthenticated: it is intended for internal use behind a corporate firewall and does not attempt to validate the caller. While effort is made to sanitize inputs, and no input is passed raw to the underlying database, a denial of service attack against VOXI itself intending to crash it or consume its resources would likely be effective. Ensure that all access to this tool is done by trusted applications and users.

## Configuring VOXI

VOXI itself has few configurable elements, most centered around how VOXI is communicated with and the database it uses. Most of the logic for VSys One and GMS 6 is defined within the respective application, including how various web aspects are handled and displayed.

### Installation

VOXI can be run as a standard desktop application or as a Windows service. As a service there is no user interface, but it can be started with no user interaction and does not require someone to have logged in. In either case, VOXI must be configured using VSys One, this defines the database connections and registration information.

To run VOXI as a desktop application, just run the application `VOXI.exe`. This will start it and launch the UI window showing its status and [Close] button.

To install VOXI as a service, from the command line, type:

```
VOXI /install
```

Use the Windows service manager to change the startup method, login authentication or other parameters as needed.

That's it. VOXI will be installed and set to automatically start with Windows. To uninstall the VOXI service, type:

```
VOXI /uninstall
```

### Database

When `VOXI.exe` is started, it connects to the default database specified in the `VSys.ini` file. VOXI will not prompt for a database to connect to. To specify that VOXI connect to a specific database connection, use the

```
database:xxx
```

command-line option, where "xxx" is the name or nickname of the database connection defined in the `VSys.ini` file.

### Port

VOXI's listening port is defined in the `VSys.ini` file in the [VOXI] section. Absent a value here, VOXI will listen on port 80.

```
[VOXI]  
Port=99
```

### Command lockouts

For cases where VOXI should only be permitted to respond to certain commands, or where specific commands must be inhibited, VOXI supports command lockouts. In the `VSys.ini` file, in the [VOXI] section, enter these commands in explicit or wildcard format.

```
[VOXI]  
validCommands=org.*,sys.*  
invalidCommands=sys.versioninfo
```

In this example, any command which begins with `org.` or `sys.` will be valid with the exception of `sys.versioninfo`. If `validCommands` is left blank then all commands are considered to be valid, except those excluded by `invalidCommands`.

### Maximum active processing engines

The maximum number of internal processing engines is defined in the `VSys.ini` file in the [VOXI] section. Absent a value here, VOXI will use a maximum of five processing engines. Values less than 1 will be assumed to be 1; greater than 100 will be assumed to be 100.

```
[VOXI]  
maxActiveProcessEngines=10
```

### Other VOXI settings

#### *Trace purging*

VOXI will purge the `voxilogs` table on a scheduled basis to keep its size down. Any log records which exceed the number of hours specified here will be removed. Absent a valid value here, one hour will be used.

```
[VOXI]
```

```
TracePurgeWindow=72
```

## **Engine retirement**

VOXI will “retire” processing engines after they’ve reached a certain age or lifetime activity. Configure the retirement behavior with two settings:

```
[VOXI]
maxEngineLife=10
maxEngineOperations=5
```

MaxEngineLife is the maximum permitted lifetime, in minutes, of an engine; MaxEngineOperations is the maximum permitted number of requests that an engine may have processed.

## **Overhead logging**

```
[VOXI]
overheadLogging=1
```

Enables the posting of diagnostic timing records to the `voxilogs` table. This is an expensive operation (time) as each request may generate dozens of `voxilogs` records, and uses up a substantial amount of disk space.

## **Command logging**

```
[VOXI]
diskLogging=1
```

By default VOXI logs all commands and their results to the `voxilogs` table. It can also log complete requests and the results from those and will do so if `diskLogging=2`. Set `diskLogging=-1` to prevent all such logs; this is helpful for performance reasons but limits diagnostic abilities using the VOXI logs monitoring tool in VSys One.

## **Other**

```
[VOXI]
testmode=1
```

Used for diagnostics with the VOXI Hammer: causes VOXI to attempt to return the same result for a request as it did in the past, even if that information is out of date, and limits the precision of various returned values. Not intended for use in a production environment.

```
diskLogging=1
```

Logs various thread creation and errors to a disk file.

```
FindJobsDefaultMaxCount=xxx
```

For the command `jobs.find`, if no `MaxCount` parameter is provided, VOXI will return no more than this number of matching job openings.

```
NoLocalization=1
```

Disables the localization tools, has the same effect as not putting the `VSys.bld` file in the same folder as `VOXI.exe`.

## **Monitoring**

VOXI listens by default on a second port – 8080 – to which monitor requests can be sent. Any HTTP GET or POST request returns the basic status of VOXI:

```
<VOXIMONITOR CommandVersion="0.0.0.5" Version="0.1.0.181">
  <STATUS ActiveRequests="0" Now="2011/07/16 00:31:43.971">
    <PORT>80</PORT>
    <DATABASE Name="temp2" ServerName="NexusDB" Address="Ishmael"/>
    <ENGINES Active="0" Available="0"/>
  </STATUS>
</VOXIMONITOR>
```

Requests on this port do not use any database connections nor any processing engines, and so are more likely to be responded to if VOXI itself has encountered problems.

The port on which VOXI listens for monitoring requests can be set:

```
[VOXI]
monitorPort=8888
```

The current status can also be retrieved with an HTTP GET request to `/status/` on the current primary port. Note that calls to `/status/` require a processing engine to complete; if VOXI is having problems allocating or using processing engines, a request on the monitoring port is more likely to succeed.

## VOXI GET Requests

Certain limited HTTP GET requests are supported by VOXI.

### `/status/`

Returns the same XML value as a request of any type on the monitoring port. Can limit its output to specific aspects of the status by including one or more of the following options, e.g.

[/status/running](#)

[/status/problems](#)

[/status/engines,memory](#)

Option	Description
running	All running processing engines.
abandoned	All abandoned processing engines; an engine is marked as “abandoned” if its sent a command and does not respond in a reasonable period of time. This permits the caller to continue on with life without waiting forever. If the command does eventually complete the engine will be moved to “available”.
available	Engines available for new commands.
problems	Possible problems: engines which have been running for a long time on a single command, engines whose status is indeterminate.
engines	Returns engines of all statuses: <i>running</i> , <i>available</i> , <i>abandoned</i> , and <i>problems</i> .
errors	A log of the last 1,000 errors.
connections	Count of active connections.
memory	Current memory utilization. Note that a very low efficiency is not an inefficient application: if memory is returned in a fragmented manner, it can’t be released to the operating system, but it is re-used when needed again.
config	Basic configuration parameters used by VOXI.

### `/image/favicon/`

### `/favicon.ico`

Both of these return the favicon graphic, if present, (same data as `org.favicon`) in icon format with no encoding.

### `/image/logo/`

Returns the logo (same data as `org.logo`) in JPEG or PNG format with no encoding.

### `/cache/xxx/`

The cache as used here is primarily for images embedded within messages and news items. A request for a document here returns that cached document if present.



## VOXI Requests and Results

All VOXI calls and responses are in XML format. These calls are submitted as HTTP POST commands.

### Requests

VOXI calls include one or more commands, and each command is processed sequentially. The overall call includes a required session identifier which can be text data of any type/length. Each command can optionally include a SEQ value which, if present, will be included in the response for that command.

#### Parameters to the VOXI element

Name	Required	Type	Description
Session	Optional	Text	
IP	Optional	Text	Dotted quad IP address (IP v4 only); if provided, must be the IP address of the user session and is used to enable/disable features on an IP address-by-IP address basis.
ForwardedIP	Optional	Text	If user sessions are done through a proxy, this should be the value of the X-Forwarded-For HTTP header or any other mechanism for determining the IP address of the actual user. Where by ForwardedIP and IP are provided, ForwardedIP will be used for determining feature enabling.
URL	Optional	Text	URL used to access the web application, if applicable; used for enabling/disabling features on a host-by-host basis.
Language	Optional	Text	Default language code for all commands in this request. Ignored if localization is disabled or the VSys.bld file is not present.

Using this parameter is preferred to using the command `localization.setlanguage` if all commands in this request will use the same language: VOXI will, if possible, find an existing processing engine already set to this language and re-use it. This saves the overhead of language switching.

#### Sample request:

```
<VOXI session="ABCD123" IP="72.0.154.3" URL="http://volunteers.yourorg.org/">
  <REQUEST command="sys.versioninfo" seq="123"/>
  <REQUEST command="sys.quote"/>
</VOXI>
```

#### Sample response:

```
<VOXI CommandVersion="0.0.0.4" Version="0.1.0.173">
  <RESULTS>
    <RESULT Command="SYS.VERSIONINFO" Seq="123">
      <VERSION>0.1.0.173</VERSION>
      <NOW>20110618 113640.812</NOW>
      <APPNAME>V Sys One eXchange Interface</APPNAME>
      <CACHEABLE Duration="120"/>
      <META Elapsed="6.39649300571364E-5" PerSecond="15633.6"/>
    </RESULT>
    <RESULT Command="SYS.QUOTE" Seq="">
      <QUOTE Speaker="Donald Knuth" Context="Preface to Fundamental Algorithms (1968)" Category="" Code="107">The
        process of preparing programs for a digital computer is especially attractive, not only because it can be
        economically and scientifically rewarding, but also because it can be an aesthetic experience much like
        composing poetry or music.</QUOTE>
      <META Elapsed="0.0187132241267697" PerSecond="53.4"/>
    </RESULT>
    <META Elapsed="0.0196122097441527" PerSecond="51"/>
  </RESULTS>
</VOXI>
```

Both VOXI and REQUEST elements support the optional *NoCache* parameter. When set, VOXI will not attempt to process the request from cache; it will still cache the results of the request if appropriate. Example:

```
<VOXI session="ABCD123">
  <REQUEST command="sys.versioninfo" seq="123" NoCache="1"/>
  <REQUEST command="sys.quote"/>
</VOXI>
```

In the above example, `sys.versioninfo` will not be processed from cache.

```
<VOXI session="ABCD123" NoCache="1">  
  <REQUEST command="sys.versioninfo" seq="123"/>  
  <REQUEST command="sys.quote"/>  
</VOXI>
```

In the above example neither `sys.versioninfo` nor `sys.quote` will be processed from cache.

## Results

Each *RESULT* element's format may vary depending on the command it's associated with. Common attributes and values include:

### CACHEABLE (element)

If present, means that the provided result may be cached for up to the indicated number of seconds. Caching is done by the caller, not by VOXI itself, and is optional but recommended to reduce the server's workload.

### OK (element)

If present, indicates that the command was successful. May have additional attributes, may have a *PROMPT* element within it.

### ERROR (element)

If present, indicates that the command failed. The *Error* attribute on this element will contain additional information about what went wrong.

## Metadata

### General

**org.aboutus, org.aboutusshort, org.contactinfo, org.maintitle, org.browsertitle**

#### Description

Requests the “About us”, “About us (short)”, “Contact information”, “Organization name” and “Browser title” values, respectively, defined in the VSys One/GMS 6 configuration, generally used in a website to describe the organization’s purpose and goals.

#### Returns

*RESULT* element whose text is the value requested.

```
<RESULT>&lt;p>&lt;b>Bespoke Software, Inc.&lt;/b> is a privately held software company incorporated in New York State.&lt;br/> We're located in Albany (upstate) New York. (It's a good thing we like snow.)&lt;br/> And we write some truly kick-butt code.&lt;/p> &lt;p>Our current primary product is a &lt;b>VSys One&lt;/b>, a volunteer management system for organizations of all sizes. We also have a special application for the Special Olympics, which we developed back in the early 90's, and continue to support and expand to this day.&lt;/p></RESULT>
```

**org.css**

#### Description

Requests the override CSS defined in the VSys One/GMS 6 configuration. The returned value may be blank.

#### Returns

A single *RESULT* element whose text is the value requested.

```
<RESULT>#header {&#xD;&#xA;&lt;br> position: relative;&#xD;&#xA;&lt;br> margin: 0;&#xD;&#xA;&lt;br> padding: 10px 35px;&#xD;&#xA;&lt;br> width: 940px;&#xD;&#xA;&lt;br> height: 80px;&#xD;&#xA;&lt;br> background: transparent url(..images/header-bg.png) 0 0 no-repeat;&#xD;&#xA;&lt;br>}&#xD;&#xA;&lt;br>#header .logo {&#xD;&#xA;&lt;br> position: absolute;&#xD;&#xA;&lt;br> top: 12px;&#xD;&#xA;&lt;br> left: 44px;&#xD;&#xA;&lt;br> display: block;&#xD;&#xA;&lt;br> width: 189px;&#xD;&#xA;&lt;br> height: 55px;&#xD;&#xA;&lt;br> text-align: left;&#xD;&#xA;&lt;br> outline: none;&#xD;&#xA;&lt;br>}</RESULT>
```

**org.pagecss**

#### Description

Requests the override CSS defined in the VSys One/GMS 6 configuration for individual pages.

#### Returns

One or more *RESULT* elements whose text is the value requested. If present, the element with the *Page* property of “\*” has the same value as *org.css* above, meaning that *org.pagecss* can be used in place of *org.css*.

```
<RESULT Page="about">#header {&#xD;&#xA; position: relative;&#xD;&#xA; margin: 0;&#xD;&#xA; padding: 10px 35px;&#xD;&#xA; width: 940px;&#xD;&#xA; height: 80px;&#xD;&#xA; background: transparent url(..images/special.png) 0 0 no-repeat;&#xD;&#xA;}&#xD;&#xA;</RESULT>
```

```
<RESULT Page="appentry">#header .logo {&#xD;&#xA; display: none;&#xD;&#xA;}&#xD;&#xA;</RESULT>
```

```
<RESULT Page="appslist"></RESULT>
<RESULT Page="calendar"></RESULT>
<RESULT Page="checkin"></RESULT>
<RESULT Page="hoursentry"></RESULT>
<RESULT Page="jobsign"></RESULT>
<RESULT Page="main"></RESULT>
<RESULT Page="myemail"></RESULT>
<RESULT Page="myinfo"></RESULT>
<RESULT Page="newsitem"></RESULT>
<RESULT Page="sendmsg"></RESULT>
<RESULT Page="trainsign"></RESULT>
```

**org.logo, org.favicon**

#### Description

Requests the “Header graphic” or “Favicon graphic” values, respectively.

Parameters

Name	Required	Type	Description
MaxHeight	Optional	Integer	If provided, requested graphic will be shrunk to fit within the dimensions provided
MaxWidth	Optional	Integer	here; if not, it may be shrunk according to system-preferred sizing.
MaxSize	Optional	Integer	If provided and the resulting image is greater than this number of bytes in size, VOXI will attempt to re-JPEG encode it at decreasing quality values until either the target size is reached or a lower quality bound of 40 is reached.

Returns

*RESULT* element whose text is the value requested encoded with Base64. (Note that the example below is intentionally truncated for space considerations here.)

```
<RESULT>iVBORw0KGgoAAAANSUHEUgAAACAAAAAgCAYAAABzenr0A...</RESULT>
```

**org.calendar**

See `calendar.list`.

**org.prompts**Description

Requests the user prompts displayed on various VSys Live pages as defined in the VSys One/GMS 6 configuration. (Note that only a limited subset of these is displayed in the example below.)

Returns

Multiple *RESULT* elements, one for each page defined.

```
<RESULT Screen="sendmessage">If you have a question, comment, praise or criticism, please fill out the information to the right and we will get back to you as soon as possible. Thank You!</RESULT>
```

```
<RESULT Screen="newsignup">To be a new volunteer, please fill out one of these applications; select the one most appropriate to your circumstances.</RESULT>
```

```
<RESULT Screen="updatedata">To update your account, please fill out one of these applications; select the one most appropriate to your circumstances.</RESULT>
```

```
<RESULT Tool="createaccount.subject"></RESULT>
```

**org.features**Description

Requests the status of the enabling of various features in VSys One in order to display or hide these features in the web interface.

Parameters

Name	Required	Type	Description
Entrant	Optional	Text	Person's 16-digit code.

Example

```
<REQUEST command="org.features" entrant="EUW38W898JD7YZU5"/>
```

Returns

```
<FEATURES>
  <FEATURE Code="HoursEntry" Enabled="1"/>
  <FEATURE Code="JobSignup" Enabled="1"/>
  <FEATURE Code="JobCheckin" Enabled="1"/>
  <FEATURE Code="NonEmailLogin" Enabled="0"/>
  <FEATURE Code="TrainingSignup" Enabled="1"/>
</FEATURES>
```

The feature `NonEmailLogin`, if enabled, means that a volunteer can sign in by using a VSys One kiosk ID, which is not an e-mail address. Therefore, if this is enabled, at login the web application should not validate the user ID as an e-mail address before passing it to VOXI for validation.

**sys.quote**Description

Requests a random quote from the quotes file. the user prompts displayed on various VSys Live pages as defined in the VSys One/GMS 6 configuration. (Note that only a limited subset of these is displayed in the example below.)

Parameters

Name	Required	Type	Description
Seed	Optional	Text	Acts as a seed to the randomizer used for selecting a quote. A given seed will always return the same quote for the same version of VOXI.

Returns

One *QUOTE* element containing the quote along with the speaker, context, category and unique code. All but “Code” may be blank.  
 <QUOTE Speaker="Donald Knuth" Context="Preface to Fundamental Algorithms (1968)" Category="" Code="107">The process of preparing programs for a digital computer is especially attractive, not only because it can be economically and scientifically rewarding, but also because it can be an aesthetic experience much like composing poetry or music.</QUOTE>

**System****sys.versioninfo**Description

Requests basic information about VOXI.

Returns

Version, the current date/time as observed by VOXI, and the application’s name.

```
<VERSION>0.1.0.173</VERSION>
<NOW>20110618 124553.809</NOW>
<APPNAME>V Sys One eXchange Interface</APPNAME>
```

**sys.commands**Description

Requests a list of all commands supported by the current version of VOXI.

Returns

One *COMMAND* element for each supported command. (Note that the example below is truncated for space considerations.)

```
<COMMAND Code="APP.GETDATA"/>
<COMMAND Code="APP.GETLAYOUT"/>
<COMMAND Code="APP.GETLAYOUTANDDATA"/>
<COMMAND Code="APP.GETLAYOUTS"/>
<COMMAND Code="APP.POSTDATA"/>
```

**sys.commandsupported**Description

Checks the validity of a given command. Not normally used.

Parameters

Name	Required	Type	Description
Query	Required	Text	Name of the command to be checked.

Example

```
<REQUEST command="sys.commandsupported" query="sys.commands"/>
```

Returns

```
<SUPPORTED Value="1"/>
```

**sys.optionsupported**Description

Not implemented.

## Options

sys.allfielddoptions, sys.contactflags, sys.genders, sys.groups, sys.jobpreferences, sys.languages, sys.partnertypes, sys.peopletypes, sys.skills, sys.specialneeds, sys.sports, sys.volunteersources, sys.volunteertypes, sys.addresstypes, sys.phonetypes, sys.partnertypes, sys.partnerstatuses, sys.partnerentrystatuses, sys.partnerentryroles, sys.kiosklocations, sys.assignmentstatuses, sys.interviewtypes

### Description

All of these return a list of the valid options of the given type. sys.allfielddoptions returns all options with a single call and is preferred for performance reasons if more than one option type list is needed.

### Returns

One *OPTIONS* element for each class plus an *OPTION* detail element for every possible value for that class.

```
<OPTIONS CLASS="CONTACTFLAGS">
  <OPTION Code="DNCALL" Name="Do not call"/>
  <OPTION Code="88JE5CA4ZC1YROAJ" Name="Do not date"/>
  <OPTION Code="Y7EE50ESZ62Z0JRP" Name="Do not drop"/>
  <OPTION Code="LCTTEQM9YNNMFQP6" Name="Do not eat"/>
  <OPTION Code="DNEMAIL" Name="Do not e-mail"/>
  <OPTION Code="DNMAIL" Name="Do not mail"/>
  <OPTION Code="5KE44TH960D1ZVLI" Name="Do not mutilate"/>
  <OPTION Code="EF7C126MMGKUDA7D" Name="Do not seduce"/>
  <OPTION Code="DNSMS" Name="Do not SMS/text message"/>
  <OPTION Code="NJ8CG3923R58HJCA" Name="Do not sneak up upon"/>
  <OPTION Code="I8G0W83NSL5ZBAMB" Name="Do not spindle"/>
</OPTIONS>
```

## Logs

All VOXI requests and their corresponding responses are stored in the `voxilogs` table. Applications are permitted to make `SELECT` queries on this table for debugging and analysis purposes.

Column	Type	Description
<code>Code</code>	<code>VARCHAR(32)</code>	A unique value associated with each row.
<code>Type</code>	<code>VARCHAR(1)</code>	<ul style="list-style-type: none"> <li>E – error</li> <li>Q – request</li> <li>R – result for a request</li> <li>q – command (one item in a possibly multipart request)</li> <li>r – result for a command record</li> <li>H – heartbeat</li> </ul>
<code>Parent</code>	<code>VARCHAR(32)</code>	For result records, references the <code>Code</code> value associated with its parent request.
<code>Command</code>	<code>VARCHAR(32)</code>	For request records, the first command in the request. If multiple commands are present, only the first will show here.
<code>Stamp</code>	<code>DATETIME</code>	
<code>Duration</code>	<code>FLOAT</code>	Processing time of the complete request, in fractional seconds. Only present for result records.
<code>Details</code>	<code>TEXT</code>	Raw XML of the request, response, or any error message returned by VOXI.

## VOXI Helper

The separate program `VOXIHelper.exe` is designed to ensure that VOXI itself remains running and running in a consistent state. It will start VOXI when `VOXIHelper` is started, and stop VOXI when `VOXIHelper` is terminated.

On a scheduled basis, VOXI Helper will first check that VOXI is running on the local machine. If it is not, VOXI Helper will attempt to start VOXI.

If VOXI is running, VOXI Helper will attempt to connect to VOXI and check its status. It first checks VOXI's monitor port, and if it receives a seemingly valid XML reply, it will execute the commands `sys.versioninfo` and `org.css`. If both of these succeed, the VOXI Helper will wait until the next check interval has passed and repeat the process.

If either of these fails, VOXI will write an error to its log. After the defined number of consecutive failures, VOXI Helper will attempt to shut down VOXI and then restart it.

### Configuring VOXI Helper

VOXI Helper has very few configuration options, and it does not connect to any databases. All of its configuration settings come from the `VSys.ini` file in the `VOXIHelper` section.

Name	Description
<code>VOXI</code>	The IP address on which it expects to find VOXI running; defaults to <code>127.0.0.1</code> .
<code>VOXILocation</code>	Optional; if present, specifies the path where <code>VOXI.exe</code> can be found. If not provided, VOXI Helper will look for <code>VOXI.exe</code> in the same folder as <code>VOXIHelper.exe</code> .
<code>VOXIAsService</code>	1 or 0; if 1, VOXI Helper will attempt to start VOXI as a service. Even if 0, if VOXI Helper is running as a service and VOXI is a registered service, VOXI Helper will launch VOXI as a service.
<code>MinRestartInterval</code>	If VOXI Helper is forced to restart VOXI, it will not restart VOXI again until this many seconds have elapsed. Default value here is 60 seconds.
<code>CheckInterval</code>	In seconds, how often VOXI Helper will perform diagnostics on VOXI.
<code>RestartAfterConsecutiveFailureCount</code>	After this many back-to-back VOXI errors, VOXI Helper will attempt to restart VOXI.
<code>Port</code>	The port on which VOXI Helper listens for status requests; defaults to <code>8081</code> .

### VOXI Helper as a Service

VOXI Helper can be run as an application or as a service. As a service there is no user interface, but it can be started with no user interaction and does not require someone to have logged in.

To run VOXI Helper as a desktop application, just run the application `VOXIHelper.exe`. This will start it and launch the UI window showing its status and [Close] button.

To install VOXI Helper as a service, from the command line, type:

```
VOXIHelper /install
```

Use the Windows service manager to change the startup method, login authentication or other parameters as needed.

That's it. VOXI Helper will be installed and set to automatically start with Windows. To uninstall the VOXI Helper service, type:

```
VOXIHelper /uninstall
```