

# JERICO



version 3.0

*Vocal supervision ... Message transmission ...*

**Jericho** is a modular communication software which proposes services to make your applications more communicative.

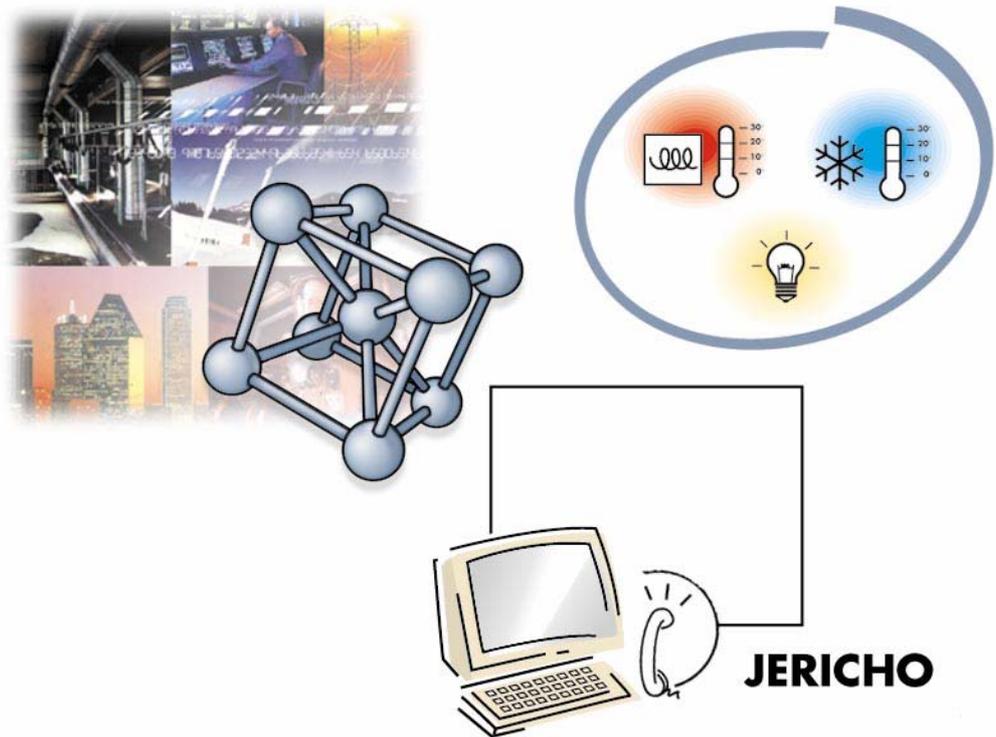
## Vocal server

**Jericho** offers to your applications a full and economic vocal interface to consult information (status or values) and send commands by telephone.

Associated with Alert, it gives a full and simple to implement solution to signal an event or a problem by telephone and to require the intervention of an operator.

## Paging service

**Jericho** proposes a set of functions to integrate in your applications some services to send written or voice messages, through various medias: telephone, short messages (SMS), pagers, beeps, fax, email, etc.



## Vocal server

The vocal server of **Jericho** provides a full vocal interface to control the working of an installation from a simple telephone handset, fixed or portable.

By using standard voice modems for the vocal communications, this functionality can be implemented in low cost conditions.

The working of the vocal server is defined by a script (Basic program), which gives a very high flexibility and a whole range of possibilities to create an efficient vocal application.

Voice messages transmitted by the vocal server can be recorded (with a microphone or by telephone) or synthesized by the integrated Text To Speech engine.



Commands transmitted from the telephonic keypad of the correspondent (DTMF codes) are decoded and interpreted by the script.



## Vocal server configuration

The working of the vocal server (welcome message, identification, vocal menus, data read and write, etc.) is defined in script programs : connection program, user programs, etc.

To write and check out these programs, **Jericho** integrates a script editor with its own debugger. The program script is a sequence of BASIC instructions that can include the main of standard basic instructions, and also a set of specific functions of the vocal server: emission of vocal messages and tones, acquisition of DTMF codes typed on the telephonic keypad, synthesis of

vocal messages, record of messages, read or write of external data, log of an event in the event journal, etc.

The vocal messages can either be recorded, locally using a microphone or by telephone, or synthesized in real time from a text (Text To Speech). To record and select the vocal messages, **Jericho** integrates a vocal recorder and a browser of vocal messages.

The management of identification code allows the selection of a vocal interface and a database context that can be specific to each operator.



## Interface with external data

To interact with external data, **Jericho** can use the real time database module **VADB**.

This module is in charge to hold a database as an image of external data required by client applications. The data exchanges between VADB and external applications (SCADA, PLC, etc.) are performed through various drivers :

- **DDE** driver to read and write data from DDE server applications.
- **OPC** driver to read and write data from OPC server applications.
- **MODBUS** driver to directly read and write data on a PLC network.
- Drivers dedicated to specific devices or SCADA.

The VADB database can be organized following a logical architecture (building, floor, office, ...). These architecture allows data to be referenced independently of their physical loca-

tion. It also allows the referenced variables to be defined relatively to a logical path that can depend on the identifier of the connected operator; it is thus possible to use the same script to control different logical entities having an identical structure (office, ...).

The configuration of the database is done through the **Jericho** user interface.

**Jericho** also can interact with external applications by using communication objects in the script:

- **DBConnection** and **DBRecorset** objects to have access to an external database by the mean of **ODBC** requests.
- **Alert** object to interact with **Alert** through its API (creation, activation, deactivation, acknowledgment of alarms, etc.).



## Applications

**Jericho** can be integrated in a software product under the form of OEM license, to give it functionalities of operator call and message transmission.

It also can be used as a stand-alone product to create communicative applications:

### ⇒ **Vocal server application**

To control an application from a distance by telephone: consultation of status (on/off, alarm, ...) or values (temperature, pressure, ...), modification of parameters (set point, threshold, ...).

**Jericho** can be used, for example, to give possibility of reactivating air conditioning or modifying the temperature set point of its office by telephone.

### ⇒ **Call server**

To trigger calls and send messages through various medias on request of client applications that are working on heterogeneous machines (Windows, Unix, ...) connected on a TCP/IP network.

### ⇒ **Call center for Alert**

To receive call from operators or users who want to signal an event or a problem and require an operator intervention. This intervention request will be transferred to **Alert** that will assume the call of concerned operators and will transmit the whole information that will have been collected by the vocal server.

This information will be able notably to include a vocal record made by the calling operator on the request of **Jericho**.



## Paging service

The paging service of **Jericho** can be solicited from an external application through its programming interface (API). This interface proposes functions to transmit messages to operators through the various medias that are managed by **Jericho**: voice messages by telephone, text messages by SMS, paging servers, fax, email, etc.

In case of unsuccessful call, the call can be automatically reiterated. If the call is always unsuccessful after a configurable number of attempts, a failure report is returned to the client application.

The voice messages transmitted by **Jericho** are stored in voice mailboxes and can be consulted by the addressee as he want during the actual call or later calls.

The paging service can also be solicited from an application that is deported over a TCP/IP network by using the **JSocket**

module. The protocol used over TCP/IP to transmit the messages can be the standard protocol of **JSocket** or any proprietary protocol (through an interpreter DLL module).

Distributed by:

