

Interactive Voice Response Script Builder Tool

Overview

The INTERACTIVE VOICE RESPONSE SCRIPT BUILDER TOOL is used to create an interactive response to a person making a call - such as record or retrieve voicemail, connect to the FAX-On-Demand service, or as a gateway to a conference – the options are almost infinite, with a capability of up to 2,400 lines per PC Chassis. It does this by creating standards based XML based script that is used by the INTERACTIVE VOICE RESPONSE SERVICE MODULE located on the CallHandler. This XML (Extensible Mark-up Language) script determines how the INTERACTIVE VOICE RESPONSE SERVICE MODULE processes any call that is passed to it from the CALL ROUTER MODULE, or via other services or modules, such as the VOICE MAIL HANDLER MODULE.

Using the INTERACTIVE VOICE RESPONSE SCRIPT BUILDER TOOL, scripts are automatically generated and the user does not need to understand the detailed structure, but simply drag the completed script to the required CallHandler directory.

The design aim of the INTERACTIVE VOICE RESPONSE SCRIPT BUILDER TOOL is to provide a simple to use, drag and drop graphical tool for use by both technical and non-technical staff for the preparation and maintenance of the XML scripts for the INTERACTIVE VOICE RESPONSE SERVICE MODULE, which can be thought of as the "glue" between service modules.

In the Call Flow Diagram shown below, an incoming call is first passed through the CALL ROUTER, which uses a look-up table to decide which module to connect the call to. A call could come in and immediately be passed to the INTERACTIVE VOICE RESPONSE SERVICE MODULE, or direct to any other service module. The INTERACTIVE VOICE RESPONSE SERVICE MODULE is capable of having a call transferred to it either directly from

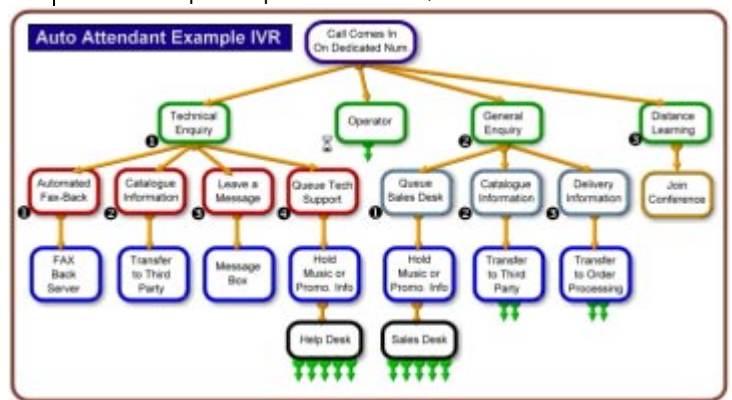
RESPONSE SCRIPT BUILDER TOOL allows the script maintainer to rapidly construct a complex voice menu.

Multiple scripts can be created and run simultaneously so that, for example, each incoming telephone line can have its own independent script.

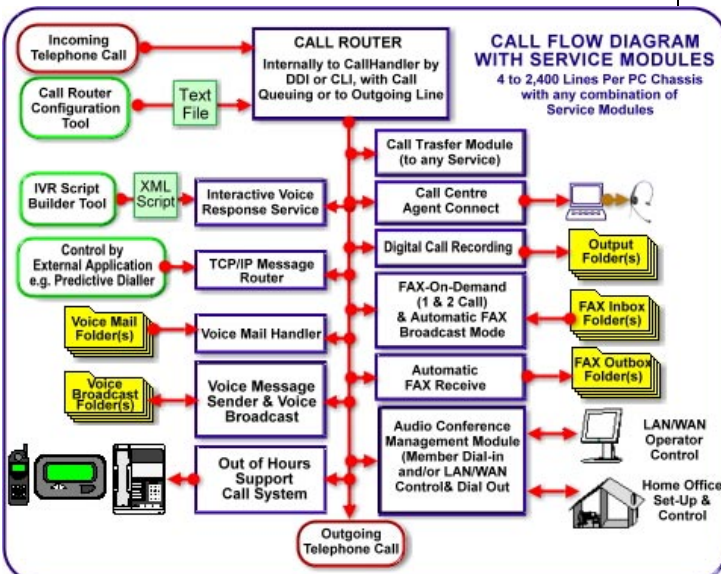
The INTERACTIVE VOICE RESPONSE SCRIPT BUILDER TOOL can be run on a desktop PC running MS Windows 95/98/NT. The XML files it creates are then transferred across the network, or via dial-up link onto the CallHandler with the system live.

Building a Typical Script - Overview

The example script outlined here, describes the Auto



Attendant call structure as shown in the diagram above. This script enables an incoming call to be directed to: the Technical section, by pressing 1, to the General Enquiry section, by pressing 2, to the Distance Learning Conference section, by pressing 3, or to an operator, where no operator input is detected. This script can be created using Chelston's INTERACTIVE VOICE RESPONSE SCRIPT BUILDER TOOL.



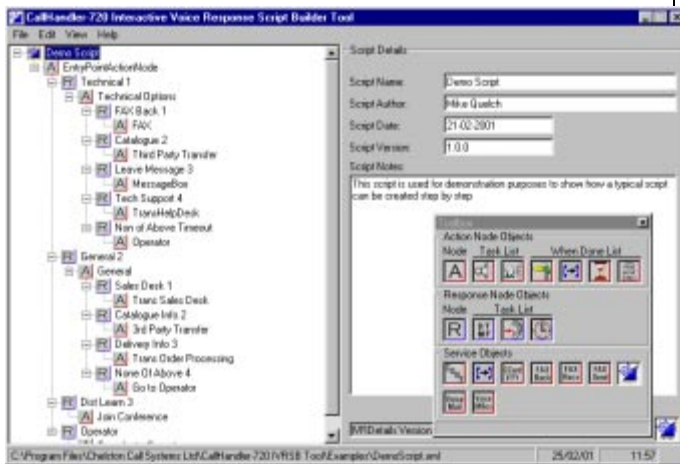
the CALL ROUTER or from any other service module. The INTERACTIVE VOICE RESPONSE SERVICE MODULE decides what to do with this call, based on the XML script. The INTERACTIVE VOICE

IVR Script Builder Tool Quick Overview

The INTERACTIVE VOICE RESPONSE SCRIPT BUILDER TOOL shown on the next page presents a graphical user interface that is split into two halves:

- ❑ The left-hand side of the tool is a window that represents a tree structure of the Interactive Response being constructed. A tree structure is built up from a series of ACTION NODES and RESPONSE NODES. Think of an Action as something that the CallHandler does, and a Response as something that the CallHandler is waiting for the Caller to do.
- ❑ The right-hand side of the window displays the current parameter details of the tree node that is presently highlighted.

The process of creating a script is first to create the graphical representation using the tree structure, which creates a sequence of ACTION and RESPONSE NODES, and then



to export this to XML format into a suitable directory. The INTERACTIVE VOICE RESPONSE SERVICE MODULE then interprets the XML script file.


All scripts start with a descriptive ROOT NODE, which comprise of the script name, date, version, author and developer notes, which may all be manually edited by the developer. (See diagram above)






An important concept is that the script is built with a combination of Action and Response Nodes, which are dragged and dropped from the floating toolbar into the script window, or into the relevant areas on the right-hand side of the tool window. Each Node can contain a mixture of Action, Response or Services, represented by plug-in modules within the system. It is these plug-in modules that actually create and perform the desired functionality within the system. These Nodes lead to other Action and Response Nodes. Objects represent Service Modules which can either be those included as standard with the CallHandler, such as play voice etc, or Custom Service Modules which can be created by the user.

Simply put, the tree structure is composed of Action Items (CallHandler does something) followed by a response by the Caller (the incoming caller does something), followed if necessary by an Action Item, followed by a Response Item, etc, etc until the desired result is obtained and the call is ultimately passed to a service which terminates the call, or it can be ended by the end call ACTION OBJECT.

The path a call takes through the Tree structure of the Script defines its interactivity. The design process requires the inclusion of all possible responses, and that ultimately the CallHandler clears the line.

ACTION OBJECTS:

Action Objects are placed within an Action Node 




-  Wait for Response from RESPONSE NODE
-  Speak Text-Based Words from input string
-  Play Audio File from disk
-  Go to other ACTION NODE
-  End Call



Available – Contact us for details










RESPONSE OBJECTS:

RESPONSE OBJECTS are placed within RESPONSE NODES.

-  Caller may press a key (triggering a DTMF detection)
-  Speech Response (triggering simple Speech Detection. E.g. "Yes", "No" etc)
-  Timeout Because no Input from Caller. May do nothing (triggering a Timeout), or does not trigger any valid response (triggering a Timeout)

SERVICE TRANSFER ACTION OBJECTS EXAMPLES:

These are Objects that take control of the Call and cause exit from the script. There is a standard Object set included with CallHandler, but custom Objects can be written in Visual Basic for specific applications.

- | | |
|--|--|
|  Verifies a Credit Card Transaction |  Enters Caller into an Audio Conference |
|  Sends a Fax to the Caller |  Performs simple Vote from Key presses into a database or text file |
|  Records the input from the Caller, e.g. Simple Voice Mailbox |  Transfer the call to another service running on the platform |
|  More Sophisticated Voice Mail System |  Sends Alert Message to specified recipient |
|  Receives a Fax from the Caller | |

Contact us to learn how this system can be applied to your application.

Typical examples include:

Build Solutions for Call Centres (traditional and virtual), Audio Conference, Voice Mail, FAX-on-demand, Fax Broadcast, Distance Learning, Web 'Call Me Back', CRM, Least Cost Routing, Calling Card, PBX Connect, IVR, Auto Attendant, Voice over IP (VOIP), Text to Speech, Rapid Application Builder for Computer Telephony, SMS Messaging, WAP, Unified Messaging, plus many custom functions. Choose pre-configured products incorporating graphical drag-and-drop configuration tools, specify a customised version for special needs, or develop your own solutions using our function rich developer packs and APIs.



Chelston Call Systems Ltd, 49 Dove Park, Chorleywood, Rickmansworth, Herts, WD3 5NY, England.
Tel: +44 01923 286501,
info@chelston.co.uk, www.chelston.co.uk