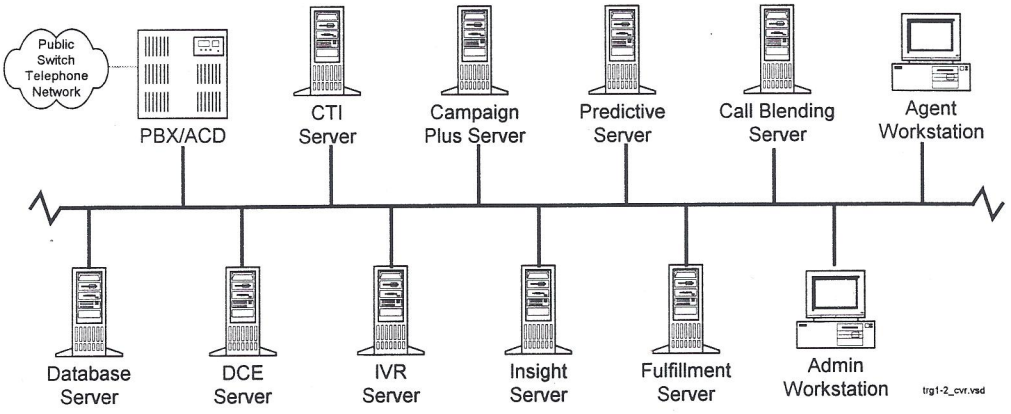
Release 1.2

VERSATILITY, INC.

TECHNICAL REFERENCE GUIDE



A Reference for Partners and Versatility, Inc.

Samuel H. Besalel
Professional Services, Versatility, Inc.
Feedback to sbesalel@versatility.com

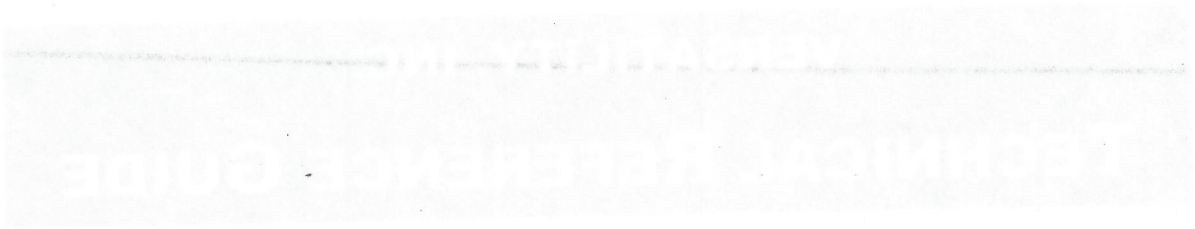


TABLE OF CONTENTS

SECTION 1 – INTRODUCTION

PURPOSE

The purpose of this Technical Reference Guide (TRG) is to provide a detailed source of information regarding Versatility Series software architecture, requirements, dependencies with other software and hardware, and other data in one centralized location. This guide is intended for the use of Partners and internal Versatility, Inc. employees. The information contained in this document is proprietary, and all existing agreements with relation to proprietary information are applicable and binding.

This is a "living document" – it is the intention of the author that this be updated regularly so that it may continue to serve as an up-to-date reference.

This guide attempts to both address its topics broadly and to serve as a roadmap to very detailed technical information. As such, many documents are cross-referenced and are included when possible.

ASSUMPTIONS

Although some attempt is made at defining operating systems, networking environments, topology, etc. while describing how they relate to the Versatility Suite, a certain level of technical knowledge is assumed. Similarly, call center/telephony industry knowledge is anticipated.

DEFINITIONS

The following definitions are provided to serve as a baseline for terminology frequently used in this document. Please note that these are defined specificially for the purposes of this Technical

Reference Guide. For a more comprehensive glossary of terminology that relates to call centers, telephony, and Versatility products, refer to Chapter 4 of the *Versatility Terminology Handbook*. Section 4 of this document contains a list of related documents and their sources.

The Versatility Series is a suite of client/server software products (detailed in *Product Descriptions* below) that support the call center environment. While a **call center** can be defined as any work group or individual that handles customer calls for sales, service, or the exchange of information, the true strength of Versatility is its integration of third-party hardware with our software to accomplish call center operations. Therefore the minimum definition of a call center *within this document* assumes more than one agent, integration of Versatility Series software, telephone switch hardware, and agent computer workstations.

The principal end-users of a call center are the individuals who use the client portion of the call center software while handling calls for the purposes of sales, service, or the exchange of information. These are sometimes known as customer service representatives, but are referred to in this document as **agents**.

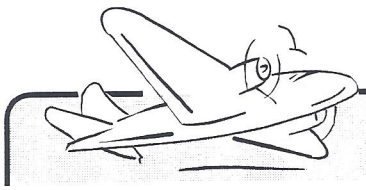
Agents typically work with a personal computer on which the relevant components of the Versatility Suite are loaded. For the purposes of this document, these desktop computers are referred to as agent **workstations**. These are in all cases IBM-compatible desktop PCs running Microsoft Windows operating system software, and should not be confused with UNIX-based hardware sometimes referred to by the same term.

There are two different definitions of workflow used in this document. **Workflow** most often refers to the flow of information between parties in order to complete a task, extending beyond a single agent . For example, if an agent needs to perform certain procedures, pass data on to a supervisor, and have the approved document returned before continuing a process (e.g., increasing a credit limit), this can be considered workflow. (For more information, see the *Versatility Customer Interactive Solutions* below.) The progression in the GUI from one screen to the next that enables call center agents to collect data associated with their task is referred to as **desktop workflow**, and is closely related to business rules, call flow and data flow, as discussed in *Initial Considerations For Call Centers* below.

Frequent mention is made in this document to the **transaction**. This refers to a Versatility call center transaction, during which an agent interacts with a customer. This is a crucial concept, and is further defined and explored in Section 3 below.

Mention is made in this document and elsewhere to the **component-based** nature of the Versatility Series. Like so many terms in the software industry, there are multiple meanings to this phrase. In reference to Versatility software, this *currently* refers to the interaction between our base product and the discrete additional shrink-wrapped and custom components that add specific functionality to a call center. For example, outbound call centers can add a predictive dialing component to their existing configuration. For the most part, these additional software components are services.

For a brief description of the various Versatility software products, see *Product Descriptions* in this document. For detailed product descriptions and their availability, see the appropriate user manual for that software title. For a discussion regarding services, see *Three General Categories* in this document.



ON THE HORIZON

In the future, these Versatility components (and third-party software products) will be granted increased access to the methods and properties embedded in our products; essentially, the business rules will be pulled out of the applications and services and will reside in middleware components.

BACKGROUND

The Versatility suite provides workflow-related services for call center agents as well as services to support call center management decisions.

The suite is component-based, with different configurations available to solve different business cases and answer varying needs. For example, some call centers take inbound calls, some make outbound calls from a predefined customer list, and others support both environments. In addition to supporting these work-

related services, Versatility also provides utilities to support marketing, strategic, and day-to-day operational management decisions. Tools are available to identify critical system issues and to assist in the integration of a diverse set of hardware, middleware and software.

SECTION 2 – VERSATILITY PRODUCTS

PRODUCT DESCRIPTIONS

The Versatility Series provides an integrated, customizable call center environment. It is the first component-based call center product, and is based on the latest technology to combine business applications functionality and integrated telephony technology with dynamic, real-time call center management facilities.

The foundation of the Versatility Suite is ***Versatility Telesales/ Teleservice (VTT)***, an application which provides call center agents with detailed customer information for sales, customer service or marketing through a familiar Windows Graphical User Interface (GUI). This application supports inbound and outbound calls, and includes the ability to customize some aspects of the GUI right out of the box.

Various components of the suite provide additional functionality for a custom solution to meet the business needs of any call center. Components of the Versatility Suite include:

- ***Versatility Administrator*** – Allows the administration of a campaign by specifying agents, customer lists, dialing methods, products and other campaign-specific data (this component comes bundled with VTT);

- ***Call Blending*** – Switches agents between campaigns and types of service (inbound and outbound) transparently based on call center client demand, helping to optimize agent workload and maximize call center productivity;

- ***Campaign Plus*** – Provides list management features for outbound campaigns and increases contact penetration using advanced record release strategies;

- ***PowerGuide*** – Enables scripting of the desktop application, call guide development and management, and desktop-level integration with other services (usually legacy systems); • *Predictive* – Establishes predictive dialing for an outbound environment;

- ***Insight*** – Provides a set of metric tools to evaluate agent performance, and to provide campaign-specific graphs and charts to assist in campaign, market, and call center operation evaluation;

- ***Fulfillment*** – Creates and/or distributes fulfillment literature from the agent's desktop using management's choice of methods (including automatic faxing, printing, and emailing of items) with;

- ***IVR*** – Retrieves call information from an Interactive Voice Response (IVR) unit and various call center components, reducing agent talk time and increasing productivity by determining in advance the purpose of the call and passing required data to the agent with the call;

- ***OpenWeb*** – Integrates a company's home page with its call center by enabling potential customers to generate an agent record for a callback at a customer-specified time, and enabling the agent to know what pages of that web site the customer viewed; and

- ***OpenTel*** – Enables the integration of the agent's phone system with the call center computer system to communicate with the phone switch and pass calls and information around the network.

These products are discussed throughout the document. For more information about product dependencies, supported databases and switches, and information as to which products were generally available (GA) at time of publication, see the *Versatility Product Compatibility Matrix* referenced in Section 4 of this document. For more detailed information on each product, see the respective user manual (other than *OpenTel*, each GA product has a user manual).

PRODUCT CATEGORIES

Counting the Software Categories

There are several different ways of classifying or grouping Versatility software; some are inclusive and others are mutually exclusive. Provided below are several different explanations of how to look at our products which can be used to communicate with clients.

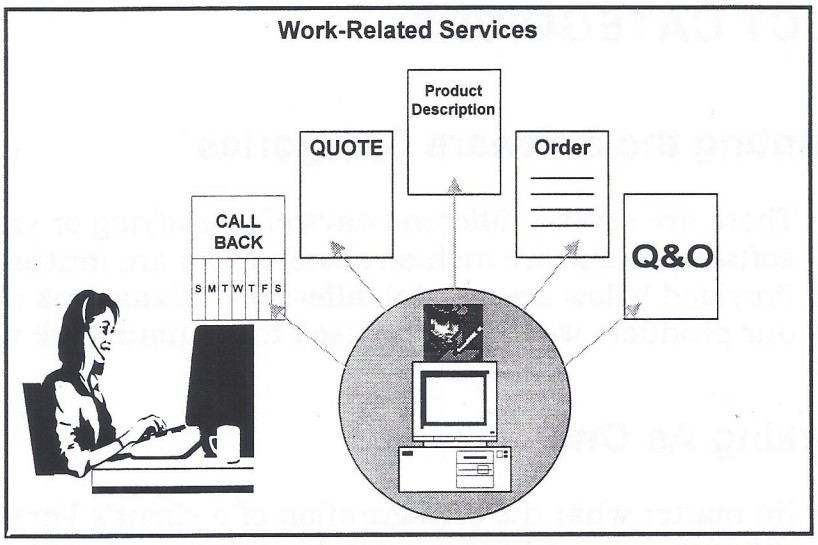
Working As One

No matter what the configuration of a client's Versatility Series solution, a key factor to consider is that our products are made to work together for a successful call center implementation. These products were developed in a client/server environment to share the workload between desktop PCs and processor-intensive tasks. Like internet browsers (a familiar analogy to many business users), the strength in this environment is our foundation in open systems. Versatility supports multiple network topologies, is certified to work with at least three popular RDMBS packages seamlessly, and supports multiple database server platforms. Our *OpenTel* telephony enabler technology works with open, published APIs, allowing functionality with multiple (and increasing numbers of) proprietary telephone switches.

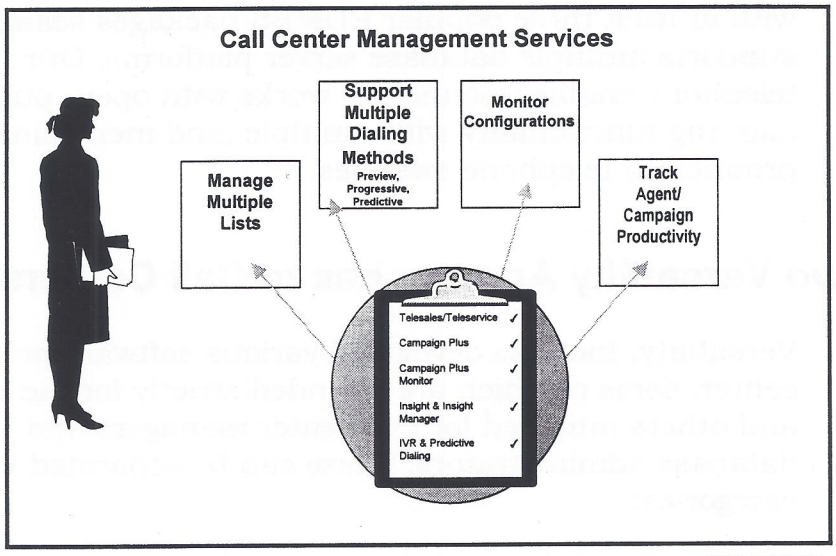
Two Versatility Approaches to Call Centers

Versatility, Inc. has developed various software products for the call center, some of which are intended strictly for the call center agent, and others intended for call center managers and network or database administrators. These can be separated into two categories:

Work-Related Services – Call center agents, using *Telesales/Teleservice*, have access to a mature call center software product designed specifically to support their needs. Providing the agent with product descriptions, quote and ordering capability, responses to frequently raised questions or objections, the ability to schedule personal call-backs, and customer history, most work tasks associated with sales, customer service, or information gathering can be accomplished.



Call Center Management Services – Additionally, many services required to manage a call center – list management, support for progressive, predictive and preview dialing, productivity reporting, and configuration monitoring and testing tools, are provided by the Versatility Series.



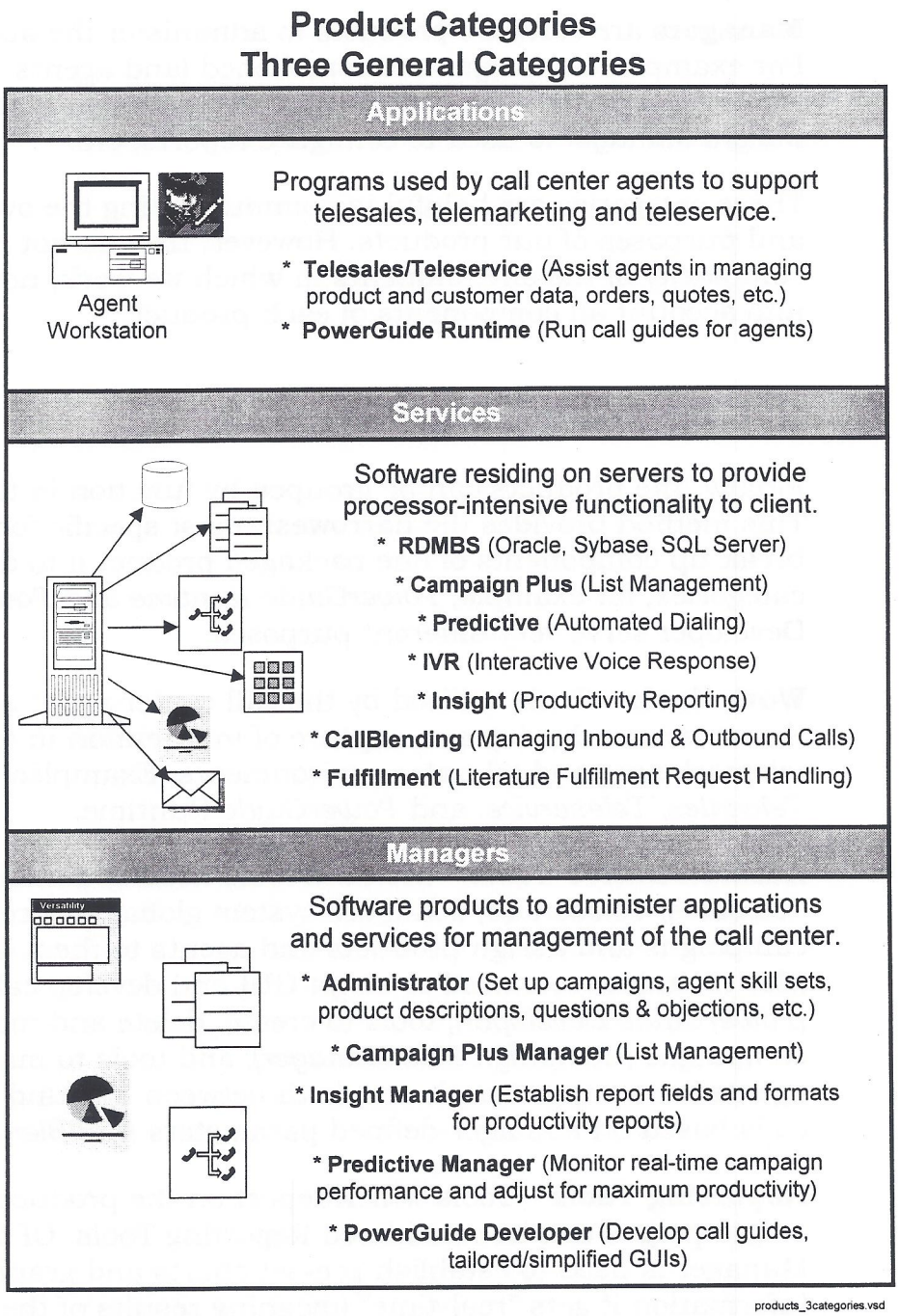
Three General Categories

Components of the Versatility Suite can be grouped into three software categories: Applications, Services, and Managers.

Applications such as *Telesales/Teleservice* are client-side programs used by call center agents to support telesales, telemarketing and teleservice. *Telesales/Teleservice* may run in

conjunction with *PowerGuide* Runtime, and can be used for scripting as well as to build custom components that are not supported in *Telesales/Teleservice*. Additionally, *PowerGuide* can manipulate the controls of *Telesales/Teleservice* by manipulating its methods and properties.

Services are server software that provide processor-intensive functionality to the client, such as providing the muscle for large



database searches and returning answers to the client.. Processing customer lists, enabling predictive dialing, developing real-time reports for agent and campaign productivity, dynamically assigning inbound and outbound calls to available agents, automating literature fulfillment, and communicating with IVR hardware are all functions provided by Versatility services. These services are *Campaign Plus, Predictive, Insight, CallBlending, Fulfillment,* and *IVR.*

Managers are software products to administer the above services. For example, campaigns are established (and agents assigned) with *Administrator*; lists are controlled with *Campaign Plus Manager; Insight Manager* is used to configure reports; etc.

These categories are helpful in communicating the overall functions and purposes of our products. However, they do not reflect the complexity of the environments in which we work, nor do they take into account all components of each product.

Specific Software Categories

Finally, the products can be grouped by function in the call center. This method provides the narrowest, most specific focus, and may break up components of one packaged product into different categories; for example, *PowerGuide Runtime* and *PowerGuide* Developer serve very different purposes.

Work Tools – Tools utilized by the call center agent to accomplish the on-screen display and capture of information in telesales, telemarketing and telesales environments. Examples: *Telesales/Teleservice,* and *PowerGuide* Runtime.

Administrative Tools – Includes tools used to administer *Telesales/Teleservice,* configure system global services, establish campaigns and assign products and agents to them *(Administrator).* Also includes tools used to script GUI and develop call guides *(PowerGuide Developer),* tools to create, delete and modify lists and campaigns *(Campaign Plus Manager),* and tools to manage how agents are swapped back and forth between inbound and outbound calls based on manager-defined parameters *(CallBlending Manager).*

Reporting Tools – Tools which report on the productivity of agents or campaigns can be considered Reporting Tools. Of these, Insight Manager is used to establish pre-set charts and graphs with information it gets "real-time" (meaning results of the current day)

Product Categories
Specific Software Categories

Work Tools	
* Telesales/Teleservice * PowerGuide Runtime * OpenWeb	Used by call center agents or prospective clients to capture or display information for telesales, telemarketing and teleservice.
Administrative Tools	
* Administrator * PowerGuide Developer * Campaign Plus Manager * CallBlending	Tools to administer the call center; create call guides; manage lists; control call flow.
Reporting Tools	
* Insight Manager * Predictive Manager	Tools which report on the productivity of agents or campaigns.
System Configuration Utilities	
* Versatility Series Configuration	Utilities which assist in the proper upload of services during installation.
System Utilities	
* Service Control Manager * Directory Cleaner	Software which provides useful functions for software loading or trouble-shooting.
Diagnostic Utilities and Simulators	
* Telesales/Teleservice Debug Logging * OpenTel Debug Logging	Software for installing, testing, configuring components of the Versatility Series.

products_specific_categories.vsd

and historically (which it draws from the campaign's historical database). *Predictive Manager* allows real-time examination of current-day predictive dialing performance, so that (at

management's discretion) predicitive dialing can be modified and reconfigured for better efficiency.

System Configuration Utilities – Utilities such as the Versatility Services Configuration, available on the Versatility Services installation CD-ROM, which assists in the proper upload of services during installation.

System Utilities – Software such as the Service Control Manager and Directory Cleaner which provides useful functions to the software loading or trouble-shooting individual at a call center site.

Diagnostic Utilities and Simulators – Versatility, Inc. has developed various software products for installing, testing, and configuring components of the Versatility Series. Many of these are not mentioned in a majority of our marketing material. There is a suite of soon-to-be-supported utilities that are currently in a quality assurance phase; other such products include the Debug Logging features of both *Telesales/Teleservice* and *OpenTel.* These are intended for the maintenance of the system and should be used with appopriate care.

Functional Categories

Another way to consider the Versatility suite is to view them via functional categories.

Customer Interaction Applications – There are three components of the Versatility suite specifically designed to interact directly with the customer. These are *Telesales/Teleservice*, *PowerGuide*, and *Administrator.*

Marketing Management Services: *Campaign Plus*, *Insight, Fulfillment,* and *OpenWeb* are all Versatility services that provide marketing management, adding functionality to the customer interaction applications by providing list management, productivity measurement, literature fulfillment, and internet-based call center connectivity, respectively.

Open Telephony Services – The Versatility Suite includes four products, *OpenTel, Predictive, IVR* and *CallBlending. OpenTel* technology is included within the applications to add CTI telephony services including predictive dialing, interactive voice recognition, and blending of inbound and outbound campaigns.

VERSATILITY ARCHITECTURE

Client/Server

The Versatility Series is based on client/server architecture and designed specifically to support call center environments. The architecture supports true client/server functionality where the desktop application administers the presentation layer, and both the client and the server manage the business rules layers.

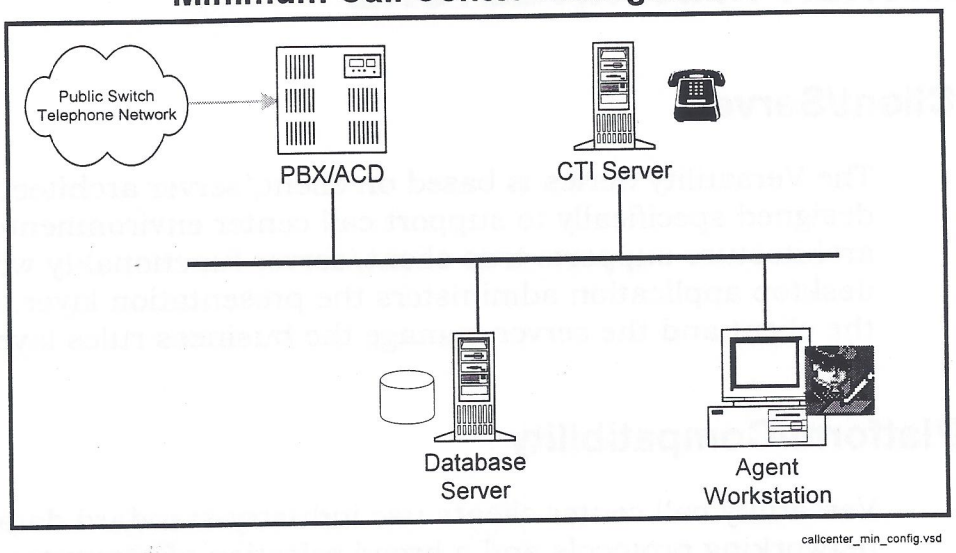
Platform/Compatibility

Versatility call center clients use industry-standard databases, networking protocols and a broad selection of computer telephony software and hardware. The agent desktop workstation is typically an Intel-based PC running a 32-bit Microsoft operating system (Windows 95 or Windows NT 3.51 or later).

Where 32-bit platforms are unavailable (typically outside of the United States), Versatility call center solutions for workstations running 16-bit (Windows 3.1 or Windows 3.11) operating systems are available and continue to be supported by Versatility, Inc. Requirements are different for these implementations, and some newer functions of Versatility may not be available. With that in mind, this document primarily addresses 32-bit implementations. The dependencies, caveats, etc. for 16-bit systems, when different from those of 32-bit systems, are mentioned herein. For further information, see the *Versatility Product Compatibility Matrix* referenced in Section 5.

At minimum, a Versatility call center solution requires *Versatility Telesales/Teleservice (VTT)* and a relational database management system (RDBMS). The typical Versatility Series solution utilizes an RDBMS server, and a telephony server. The database server may be any processor and operating system that supports TCP/IP networking and one of the American National Standards Institute (ANSI) Structured Query Language (SQL)-compliant databases discussed below. The telephony server is typically a Windows NT machine which can monitor and manage thousands of concurrent agents and telephone extensions in many campaigns and different Versatility services.

Minimum Call Center Configuration



The Versatility Series includes native support for Oracle, Sybase, and Microsoft SQL Server RDBMSs. Regardless of the database of choice, the Versatility Series will provide application and administrative access, via SQL statements, to data residing on a variety of platforms, local or remote.

VERSATILITY CUSTOMER INTERACTIVE SOLUTIONS

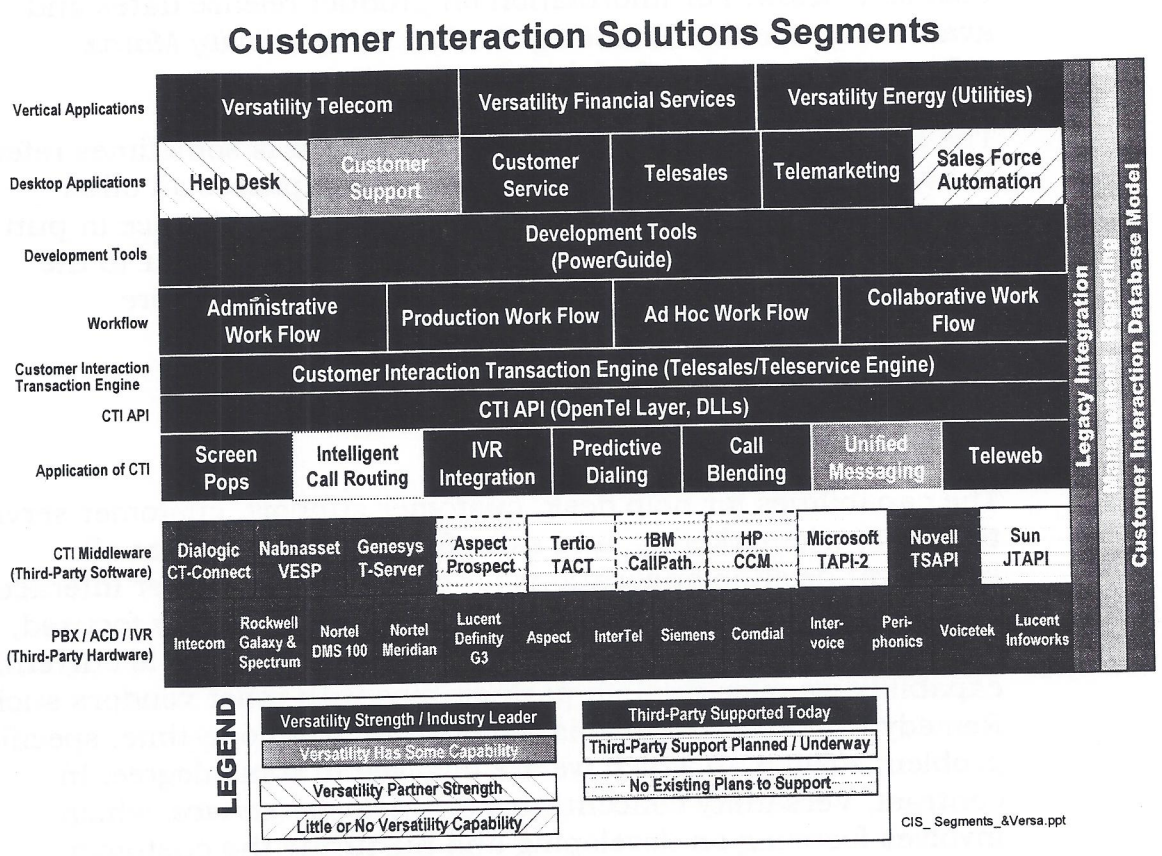
Architecture Within the CTI/Call Center marketplace

The diagram below depicts the different layers of the Versatility Suite within the context of the CTI/call center environment.

The layers addressed are vertical applications, desktop applications, development tools, workflow, customer interaction transaction engine, CTI API, the application of CTI, CTI middleware, and connection (switch, ACD, and IVR) hardware. They are discussed in this order, following the graphic.

Vertical Applications

Telesales/Teleservice has over 50 screens and about 40 additional dialogue boxes to help agents with customer service, sales and marketing activities in a call center. The different functions built into this application are inherently applicable to many businesses.



Additionally, some aspects of the GUI are easily changed, such as menu items, window titles, etc., providing further flexibility and versatility to various types of commercial markets.

Versatility also offers different software configurations to support identifiable vertical markets, each with additional industry-specific tools and focus. To date, three core "verticals" – telecomunications, financial services, and utilities/energy – have been identified. Versatility has developed specialized call center solutions for each vertical market.

For example, the *Versatility Telecom* (VT) product incorporates a telecommunications-specialized GUI for *Telesales/Teleservice,* providing similar functionality as the VTT component found in the other verticals. It also contains specific components adding telecommunications-focused features and functionality which were designed to empower Telecom customer service agents to fully meet the needs of that carrier's customer base.

The other two core vertical markets identified to date benefit from features specific to those industries. For more information on vertical configurations, refer to *Versatility Customer Interactive*

Solutions below. For information on product release dates and availability, see the *Versatility Product Compatibility Matrix* referenced in Section 5 of this document.

The standard version of *Telesales/Teleservice* is sometimes referred to as the *Commercial* vertical, since it encompasses all other commercial markets. For implementation-level guidance in putting uncustomized commercial VTT solutions in place, refer to the *Versatility Implementation Guide* (see Section 5 for more information).

Desktop Applications

The capabilities for help desk, customer support, customer service, telesales, telemarketing, and sales force automation are all important desktop applications for call center customer interaction solutions. *Help desk* applications, which are internally focused, include actual hardware/software configuration. This is currently a capability we offer through partnerships with other vendors such as Remedy. *Customer support* is intended to mean one-time, specific problem solving, to which we are involved to some degree. In contrast, Versatility concentrates on *customer service*, which involves focusing on developing and nurturing the customer relationship. This type of solution is geared toward the customer life cycle: acquisition, service, sales, and retention. For example, when a customer calls into the call center asking for product assistance, a billing inquiry, rate plan explanation or a new service plan, call center agents must be able to fully understand the customer's needs and be familiar with the current products and services the customer uses. In this way, the agent can offer the solution that will, in the long run, extend their relationship with the customer or vendor. Any customer is a valuable resource, and focusing on the relationship is in the long run better for all involved.

Of course, looking at the next areas on the application layer of the diagram, our flagship product offers solid *telesales* and *telemarketing* capability. Finally, *sales force automation* – in terms of supporting a mobile, remote sales force while on the road – is the one area of servicing the call center environment in which we currently have no entry or specialized desktop application, as this has not been a demand of our customers and prospects. When this type of functionality is required, partnerships can be investigated further.

Development Tools

The development tool layer is satisfied by our *PowerGuide* product. This application provides extensive scripting using a Visual Basic backbone to allow seamless control of thousands of methods and properties. This enables call center managers to provide complete scripting of the desktop agent application using a GUI front-end tool with little or no programming required. This layer also supports integration to back-end operational support systems or third-party applications through OLE Automation, DDE (for 16-bit systems), emulation packages, and other means. Tight integration to disparate systems is usually a requirement of Versatility Series implementations.

Workflow

Both *Telesales/Teleservice* and *PowerGuide* provide for workflow. Workflow is the process of the agent or manager initiating an event – one task or a series of tasks – to achieve resolution for a customer service situation(a "case"). Often, the tasks involve other entities needed to solve the case.

Workflow can also be used to disseminate information from administrators to call center agents. For example, if there is a new product being introduced into a call center, the manager can create a workflow task that requires agents to review materials describing the new product prior to logging on to the system to make or receive calls.

Workflow is strongly dependent on pre-defined business rules. Tasks cannot be accomplished without the defining business rules for the agent and any other entities. Versatility workflow tools include the ability to "push" documents regarding the case to the other entities; this routing can be accomplished serially, in parallel, or both. Workflow within Versatility is agent-centric (all tasks and subtasks that comprise the event are initiated by the agent).

Templates for each vertical will include some "canned" workflows to solve cases specific to each respective vertical market. For example, a billing adjustment template supports modifying a customer bill in *Versatility Telecom*, and a loan application approval template can be used with *Versatility Financial.*

The imminent first generation of workflow components includes the ability to automatically generate email notification at certain stages

of the case. For example, in the case of a billing dispute, the agent who first logs the case may follow a workflow which passes the case to the billing department for resolution. If the billing department resolves the case, the workflow can include an automatic email notification to the originating agent, who can then make a callback to notify the customer that the situation has been resolved. The workflow may also allow for escalation to a supervisor should the situation not be resolved at that level.

Customer Interaction Transaction Engine

This component tracks all call information that occurs during customer contacts, including call dispositions, Fulfillment requests, scheduled callbacks, call durations, product interests and sales, and agent activitity. The information tracked is often used to populate a centralized data warehouse for tracking and analysis and for creation of future sales, marketing and service campaigns.

CTI API

The Application Program Interface (API) in the Computer-Telephony Integration (CTI) environment is key to ensuring the proper handling of information between components of the call center. Versatility provides *OpenTel* as a CTI enabler to provide the proper method of native communication between our software solutions and a variety of telecom switch hardware and their telephony enablers.

Application of CTI

It is the application of CTI that: enables screenpop through ANI and DNIS, provides for intelligent call routing through the appropriate hardware, permits IVR integration, allows progressive, preview and timed preview, and predictive dialing, allows for hands-free telephony, utilizes additional utilities to provide call blending of inbound and outbound campaigns, provides the framework for unified messaging, and allows management of call center operations over the world-wide web. These functions are all possible with the Versatility solution.

CTI Middleware

There is a diverse amount of third-party middleware – software that ties together hardware components and data systems, thus enabling our software products to communicate with switches and IVRs.

Middleware is often referred to as the "glue" that binds together the computer and telephony hardware and the desktop applications that enable call centers to coordinate and manage the actual process of contacting or being contacted by existing clients or potential customers.

Connection Hardware

PBX switches, ACD units and IVR units are the largest initial expenses for a call center, and subsequently there are various companies that manufacture and support this hardware. These components are required to enable advance telephony capabilities in a call center. Depicted in the above graphic are a representative (not all-inclusive) sample of the companies and equipment we support.

SECTION 3 – THE VERSATILITY TRANSACTION

THE TRANSACTION

Overview

The Versatility *transaction* – an event associated with a customer record – is the heart of Versatility's agent-focused call center software. When a call is initiated between an agent and a customer (or prospect), a transaction is started.

Scope

Although transactions *can* be controlled outside of *Versatility Telesales/Teleservice*, for the purpose of this discussion, the term 'transaction' is used assuming the reader *is using VTT*. It is not relevant whether the version of *VTT* is the stand-alone product or a version of *VTT* for a specific vertical market.

Definition

A transaction is a measurable and relevant interaction between an identifiable customer (or prospect) and a single agent.

Each portion of this definition is important to the understanding of the transaction and how it relates to our products.

Transactions must be *measurable* in terms of quantity (how many transactions occurred), length (what is the duration of the call and the post-call processing), and outcome (was the

objective of the call obtained). *Versatility Insight* can measure all aspects of a transaction.

Each transaction should be *relevant* – considered important to either the customer or the agent – for several reasons. For one thing, transactions that are irrelevant drain call center resources that could be otherwise focused on the call center's primary business purposes (e.g., making sales, or responding to customer calls). Relevance is also an issue because, if the interaction were not relevant, the results of the call center's productivity tracking (if any) would be invalid.

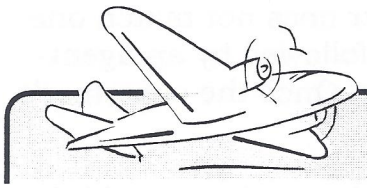
Relevance is primarily controlled via management of the specific campaign; the call (whether inbound or outbound) must be related to a specific campaign in order for the transaction to be relevant. Examples of irrelevant transactions include any predictive calls that were abandoned when no agents were available (either before or after a customer answered), or inbound calls with no identifiable customer (see below).

The transaction must involve an *identifiable* customer or prospect for several reasons. In order for call centers to comply with federal legislation, a customer or prospect must be identifiable in the event that individual requests not to be contacted (so that they can be put on a "stop" or "don't contact" list). For tracking purposes, the call should be identifiable so that it can be ascertained that the transaction (and therefore the statistics regarding the call) is valid. Based on the goals of the campaign and desires of call center management, there may be a requirement to follow up on certain contacts (for example, to make a call-back to a "maybe" outcome, or to ensure satisfaction on the customer's part after a sale).

The *customer* or *prospect* includes not only that individual but also the information about that individual. For example, for the purposes of maintenance, a call center supervisor may need to access and make changes to a customer account – say, to enter a customer's change of address that was received in the mail. In this case, the customer would not need to be contacted, but access must be granted to that customer's information. This would take place during a transaction.

A transaction involves a *single* agent. If a transaction requires a transfer to another agent, it is tracked as a separate transaction, to reduce complexity and provide accurate reporting statistics.

Versatility Predictive can be considered an agent in the definition above. If a contact is achieved in an outbound, predictive dialing environment, the call is transferred to an agent and, for all intents and purposes, the transaction begins. However, for all calls for which contact is *not* an outcome, *Predictive* (in concert with Campaign Plus) essentially statuses the transaction with a call outcome on behalf of the agent (writing to the Campaign Plus call history file), and causes calls to be recycled.



ON THE HORIZON

In a future release, upon a "no contact" outcome,*Predictive* will also update the CALL_HISTORY table when it statuses the transaction.

Components of the Transaction

There are three main components of the transaction:

- Initiating a Transaction
- Interacting with the Customer
- Statusing the Transaction

Three Components of a Transaction

Initiating a Transaction	*Interacting with Customer*	*Post-Transaction Processing*
Methods of Getting a Customer · List Manager (e.g., *Campaign Plus*) · Versatility List or Callback list · Agent-initiated Customer Search · Customer Add · Inbound ANI screen-pop	**Solving needs during the call** · Create Quotes/Orders · Generate literature fulfillment · Refer caller to other dept./supervisor · Update customer info · Record customer's products of interest · View call history · Create system or personal callbacks	**Updating Customer History** **Statusing a Transaction** · **Call Outcome** · **Call Result** · **Result Reason** **Perform Exit Transaction**

components_of_transaction2.vsd

Initiating a Transaction – Inbound

Initiating a transaction involves obtaining a customer (or prospect) with whom the agent interacts. This can be substantially different in inbound and outbound environments.

For example, in an inbound environment, a call is routed to the agent from the PBX. If the call arrives at the agent along with ANI information (sent via a different path), and the agent is in the customer database, there may be a screen pop, showing detailed customer information.

In the same situation, the customer's call may arrive at the agent's workstation without a screen pop – either because ANI was not sent, or the caller's number does not match one in the customer database. This can be followed by an agent-initiated search of a customer database. Once the customer's record is found, the transaction begins.

If the caller is not already a customer, the agent may add that customer by filling out a new customer profile. At minimum, the customer's first and last name, area code and phone number are required. The agent should attempt to capture as much additional information as possible.

Initiating a Transaction – Outbound

Alternatively, in an outbound environment, there are at least three ways an agent may start a transaction.

First, the agent may get a customer record from a list manager such as *Campaign Plus*. The customer can be contacted using one of three dialing methods. For progressive dialing, each customer is dialed directly from the list, one at a time, by the agent. For preview or timed preview dialing, the agent may view the customer record for an agent-determined or pre-set amount of time before dialing the customer. Or the numbers can be dialed from a predictive server, with the customer transferred to the agent upon contact.

Second, an agent in an outbound call center can reach a customer with a personal callback. This is a scheduled follow-up call to a previously contacted customer. In a sales environment, perhaps the customer needed more time to consider his purchase, or wanted to consider literature sent

by the agent prior to a decision. In a service environment, perhaps the agent scheduled the callback to ensure the customer had satisfactorily resolved a problem previously discussed with the agent.

Finally, the outbound agent can start a transaction by performing a customer search on the list available to her campaign. One reason for using this method is a record that was flagged by an agent for maintenance.

Interacting with the Customer

There are many functions an agent can perform while interacting with a customer during a transaction, and also some things not permitted. In general, an agent can update customer information; create personal or system callbacks; create one or more quotes; convert a quote to an order; create one or more orders from scratch; view a customer's call history; or satisfy the reason for the call and end the call part of the transaction. (It is not the intention of this document to discuss the full functionality of the Versatility call center application; refer to the *Telesales/Teleservice User Manual* for specific, detailed information on this topic.)

Things that an agent *cannot do while* interacting with a customer during a transaction include exiting *Telesales/Teleservice*, interacting with more than one customer at a time or conducting more than one transaction at a time, or switching campaigns after creating an order, quote, request for fulfillment, product of interest or a referral.

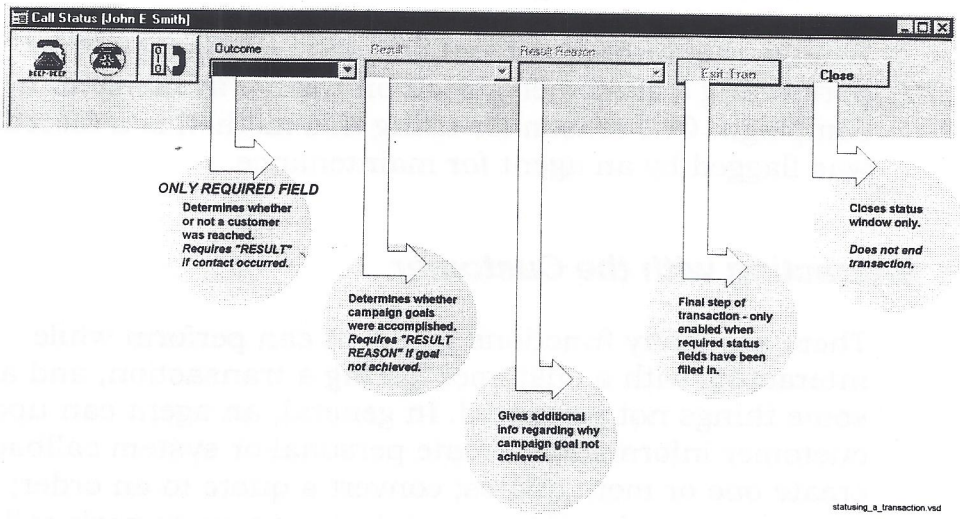
Post-Call Processing

After the agent has completed his work on the telephone, the call is ended. However, whether or not the goal of the campaign has been achieved, the transaction itself is not complete until the agent finishes the post-call processing. For *Telesales/Teleservice*, this is referred to as "statusing the call," which is performed in the call status window.

Elements Required to Status a Call

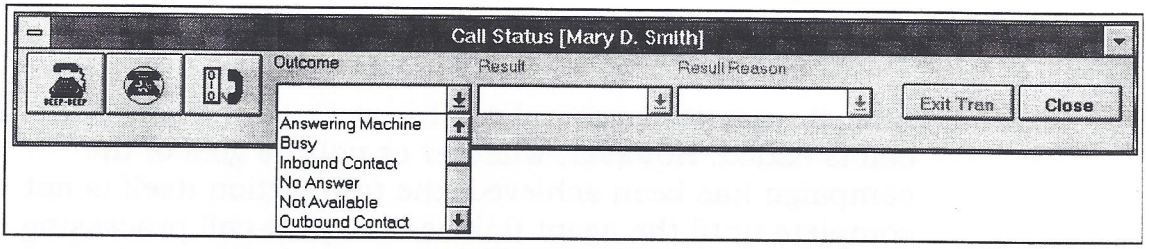
There are three fields in the call status window, each of which describes the call: *Outcome, Result,* and *Result Reason.*

Statusing a Transaction



Call Outcome

Without exception, a call *outcome* is required for each transaction. The outcome describes what happened during the attempt by the agent (or predictive service) to reach a customer. The most desireable outcome, of course, is *Contact.* The list of possible outcomes is established by the call center administrator and is customized for each campaign. The next three most common outcomes *(Busy, No Answer, Answering Machine)* are represented by the buttons to the left of the status fields in the call status window. (These can be removed in the Administrator mode if desired.)



Some typical outcomes include:

Inbound Contact — When the agent receives a call from a customer. Selecting this outcome ordinarily forces the

agent to complete the *Result* field before ending the transaction.

Outbound Contact — When the agent reaches a customer and begins a conversation. Selecting this outcome ordinarily forces the agent to complete the *Result* field before ending the transaction.

Busy — During an outbound call, busy describes the outcome when the agent (or predictive service) reaches a standard busy signal for that call. When this outcome is selected, *Telesales/Teleservice* automatically recycles the customer record so that it will be dialed again at a later time.

No Answer — During an outbound call, this describes the condition when the agent (or predictive service) gets no answer. When this outcome is selected, *Telesales/Teleservice* automatically recycles the customer record so that it will be dialed again at a later time.

Answering Machine —When an answering machine is reached during an outbound call. When this outcome is selected, *Telesales/Teleservice* automatically recycles the customer record so that it will be dialed again, generally at a different time of day.

Not Available — When the customer is not available and no callback time has been established.

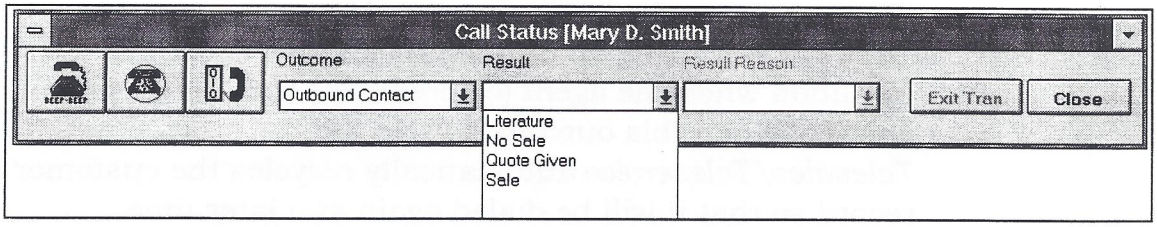
There are 128 outome codes reserved by Versatility software (refer to the *Campaign Plus* user manual for this list). Additional outcome codes can be assigned by call center management.

Any time a call center administrator chooses to implement user-defined outcomes, she runs the risk of interfering with the proper functionality of related processes. For example, *Insight* reports use outcomes, results, and result reasons to generate reports

on the productivity of a call center. Changing those status fields or their behavior may yield unexpected reporting results.

Result

A call *result* describes the final result the agent's conversation with the customer. Was a sale made? Was a quote given? Did the transaction result in no sale? Each call center has a unique set of valid result codes set up by the call center administrator. Ordinarily, the *Result* field will automatically become accessible if the *Outcome* field indicates contact was made.



Some typical results include:

Sale — A product or service specific to that campaign has been sold and an order taken. Without an order, VTT does not allow the agent to choose *Sale*. A *Result Reason* is ordinarily not required for a sale.

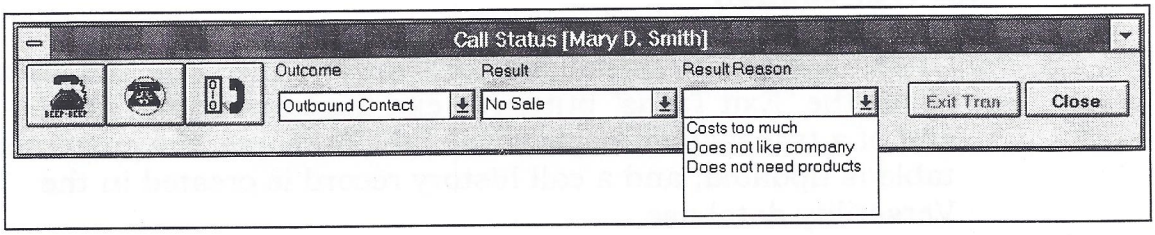
No Sale — The agent was not successful in selling the product or service to the customer. Selecting this result ordinarily forces the agent to complete the *Result Reason* field before closing the call status window.

Literature — During a call, a sale was not made but the customer will be sent literature to help in a purchase decision.

Quote Given — During the call, the agent gave a quote to the customer but no sale was made. This information stays with the customer record, aiding the agent in the event of subsequent customer contact.

Result Reason

The third status field, *Result Reason*, provides call center management with additional information as to why the result for that transaction was obtained. (It therefore is ordinarily dependent on a *contact* outcome.) This is most frequently implemented to describe why the goal of the campaign (for example, selling a product or service) was *not* achieved in that transaction. This information can be polled with *Insight* and analyzed to increase productivity.



Result Reasons are also established by the call center administrator and customized for each campaign. Some typical result reasons include:

Too Costly — The product or service costs too much and the prospective customer cannot afford it.

Not Needed — The product or service is not needed by the prospective customer.

Own Competing Product — The prospective customer already owns a competing product/service.

Most of this information can be valuable, whether to agents good at cross-selling or up-selling products or services, or to call center management to reveal trends with which they can weigh their business decisions (pricing, etc.) based on the information they glean during a campaign.

Why do the descriptions above frequently caveat that certain behavior "ordinarily" occurs in the status window? There are expected relationships between the status

fields that come pre-set by default in Versatility, but which are configurable. For example, in the *Statusing the Transaction* graphic and the descriptions above, the expected relationships are as follows: *Outcome* is always required; *Result* is required when the outcome is "contact"; and *Result Reason* is required if the campaign goal (i.e., "sale") was not achieved during the call. However, at the discretion of the call center administrator, these dependencies can be disabled (e.g., *Result Reason* not required for a negative response), or even enabled (use *Result Reason* to determine the main reason a customer decided to purchase the product).

Exit Transaction

Once the appropriate call status fields are filled out, the agent clicks the "Exit Trans" button to exit the transaction. At the end of a transaction, regardless of the outcome, the customer table is updated, and a call history record is created in the Versatility database.

All activity that occurred during the call is recorded. For example, if a quote was taken, the quote table is updated; if an order was taken, the order table is written to; if literature was requested, the literature fulfillment table is updated; if notes were taken by the agent, these are saved with the customer history.

These events occur independent of the agent. From the agent's perspective, ending the transaction sends a message (an RPC call) to the telephony enabler notifying the system that the agent is ready for another call.

SECTION 4 – TECHNICAL DATA

INITIAL CONSIDERATIONS FOR CALL CENTERS

Any comprehensive discussion of call center architecture, like any implementation of a call center solution, should focus first on the following three items: business rules, call flow, and information/data flow.

The potential complexity of a call center is vast. When considering the many choices – network structure, database requirements, hardware and software relationships and dependencies, GUI and vertical market specialization – it is easy to lose perspective. But the repercussions of failing to consider these concepts early in the process are serious. In a best-case scenario, it means not serving the customer in the most efficient manner. In the worst case, it can mean a failed implementation attempt, embarassment for the implementation team, damage to the corporate relationship, and financial ruin.

Business Rules

Considering business rules in a call center implementation in support of the business function is of foremost concern. What are the parameters of the customer's business? What are the business assumptions that define how the agent can use call center software? Let's use a hypothetical example.

CASE STUDY

ABC Software Company has an existing software product, Product 1.0, and has just released a new version, Product 2.0. It has a call center with inbound (customer service) and outbound (sales) functions.

The business rules provide the framework for how the call center will function. For example, in their inbound (customer service) call center environment, tracking previous customer calls may be considered critical for ensuring customer satisfaction. One business rule, therefore, is that support calls and their outcomes must be documented by ABC agents as customer history. When a call is received, the agent must know: Has the owner of this software called previously for support? Is the incoming call an open ticket (a second or subsequent call regarding the same question or problem), requiring a ticket or case number? Are case numbers identified with certain leading digits – 1 for "installation questions", 2 for "printing issues," 3 for "exporting files," etc.?

The business rules determine what data must be collected, setting the parameters for the transaction that is the heart of a call center. For example, If a customer has received free support for a specific software product for ninety days, are they still allowed to access the 800/888-number toll-free tech support line, or must they pay for continued support? This business rule must be established in advance in order for call center managers to provide for billing after 90 days of support, or to assign agents to particular campaigns (free support or pay support), or to decide whether to track effectiveness for that particular campaign. As another example, how many classes of customers will a call center have? Do they require different campaigns, or different call-handling priority at the switch? Any one company's set of business rules will be unique. Understanding them is of paramount importance for designing a call center solution.

Before a campaign should ever be started, the business rules must be clearly defined. Only after this occurs should data requirements be considered. What information needs to be captured? From where? What does that data represent? Why is it being collected? Where will it be used, and how does it get there? These questions can only be properly answered *after the business rules are understood.*

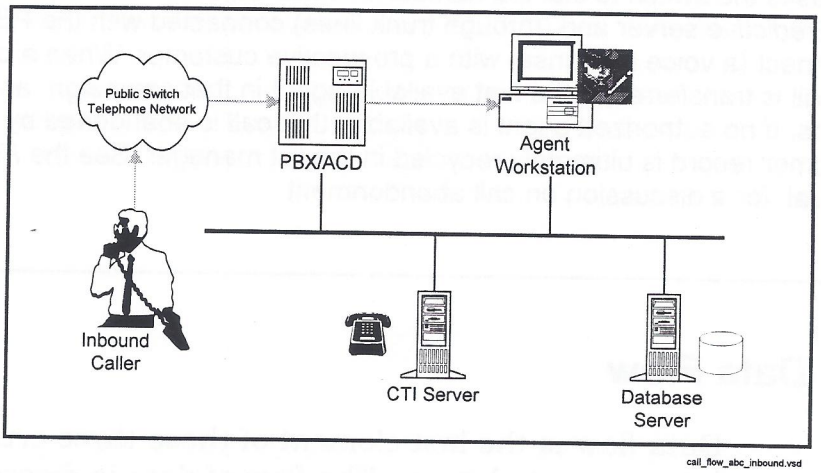
Call Flow

The call flow is essentially the movement or flow of the actual inbound or outbound call ("the voice") within the call center. It is discreet from the flow of data, which may be transferred along the computer network. Again, let's look to ABC Software Company for a simple example.

CASE STUDY

In ABC's inbound customer service call center, a call comes in from the Public Service Telephone Network to the Private Branch Exchange (PBX), which has an Automatic Call Distributor (ACD). The CTI server is notified that a call has been received.

Sample Call Flow
ABC Software Inbound Call Flow

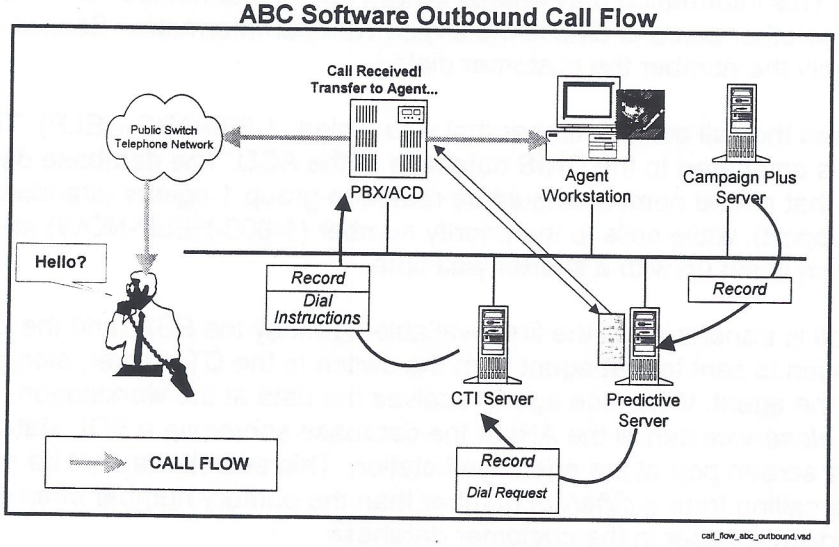


In this model, our call center implementation team played it by the book, laying out all the business rules in advance. Therefore, they have determined that they want to have more than one pool of agents to answer calls, based on the business rule that there are two classes of customers: *standard* and *priority*. The priority customers may have paid an additional amount when purchasing their software in order to achieve priority status; in turn, they call a special number (1-800-HELP-NOW) which guarantees they receive telephone support within a specified amount of time (e.g., one minute). Standard customers call the regular support number (1-800-ABC-HELP), which averages a response time of five minutes or less. These may be the same agents; however, the priority calls are handled first.

Based on the number dialed, the PBX/ACD routes the call to the first available agent, and the transaction begins.

In an outbound call center using predictive dialing (ABC Software Company's sales division), *Predictive* is passed a customer record from *Campaign Plus*. *Predictive* takes

Sample Call Flow
ABC Software Outbound Call Flow



the record and sends it (with a request for predictive dialing) to the CTI server, which in turn asks the switch to dial the number. A voice detection unit (VDU) card installed on the *Predictive* server and (through trunk lines) connected with the PBX/ACD listens for a connect (a voice response) with a prospective customer. When a connect occurs, the call is transferred to the first available agent in that campaign, and the transaction begins. If no authorized agent is available, that call is abandoned by the PBX, and that customer record is ultimately recycled in the list manager. See the *Predictive* user manual for a discussion on call abandonment.

Data Flow

Data flow is the last element of these three crucial, interconnected items. The flow of data is discreet from the flow of the call (which is controlled by the PBX/ACD), but also critical in the effective call center.

Let's look at ABC Software Company's inbound customer service call center to follow the flow of data.

CASE STUDY

An ABC customer dials the support line; the call comes from the Public Service Telephone Network to the company's PBX/ACD. The CTI server is notified that a call has been received.

At the switch, there are two pieces of discreet information captured that we are concerned with, both of which are transferred to the receiving agent. The first is Automatic Number Identification (ANI), which is the phone number the customer is calling from. This information may enable screen pop, if that number is in the customer database. The other piece of data is the Dialed Number Information Service (DNIS), which is simply the number the customer dialed.

DNIS captures the call center number that was dialed (1-800-ABC-HELP). This information is compared to the DNIS database at the ACD. The database determines that calls to that phone number should be routed to group 1 agents (standard ABC customer support), while calls to the priority number (1-800-HELP-NOW) are sent to group 2, which is set up with a shorter wait time.

Thus, the call is transferred to the first available agent by the PBX, and the DNIS and ANI information is sent to that agent from the switch to the CTI server, along the network, to the agent. When the agent receives the data at the workstation, *Telesales/Teleservice* sends the ANI to the database server via a SQL statement, requesting a screen pop at the agent workstation. This search may not be successful, if the user is calling from a different number than the primary number of an ABC Software registered user in the customer database.

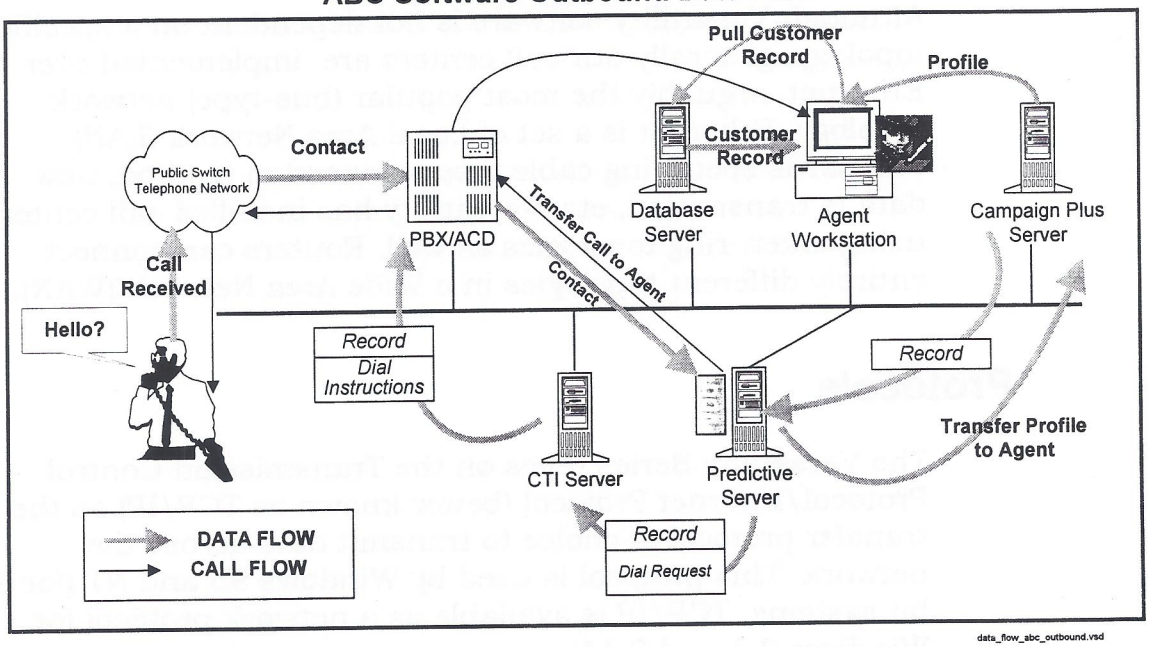
The data flow of an outbound call is very similar.

CASE STUDY

In ABC Software Company's sales division, again using predictive dialing and list management software, *Predictive* is passed a customer record from *Campaign Plus*. The *Campaign Plus* list is a much smaller subset of information that's in the customer database; it may have 10 fields where the customer database has 100. *Predictive* takes the record and sends it (with a request for predictive dialing) to the CTI server, which in turn asks the switch to dial the number. The VDU listens for a connect, and the PBX/ACD transfers the call to the first available agent in that campaign.

In this case, the potential customer is a known factor, since it came from ABC Software Company's own customer database. Therefore, when the VDU discovers a connect, it notifies *Predictive*. *Predictive* then sends a request to *Campaign Plus* to forward what information it has (the customer profile) along the network to the agent.

Sample Data Flow
ABC Software Outbound Data Flow



When the agent receives the call from the PBX/ACD and the customer profile from *Campaign Plus, Telesales/Teleservice* must pull the entire customer record from the customer database (most likely using the customer ID). In this way, the customer's history with ABC (if any) will be available to the agent, which may provide information helpful to make a sale. In the meantime, the agent has some information (the customer profile) with which to begin her sales pitch.

Desktop Workflow

Together, the business rules, call flow, and data flow make up the desktop workflow – the flow of work an agent performs at the desktop. After the three previously described elements are planned out, data requirements are determined, and a GUI can be built to incorporate the business rules and allow for the call and data flow.

Desktop workflow therefore describes the possible progression in the GUI from one screen to the next, and is limited by the business rules. For example, an agent on a call cannot end a transaction without first providing a status for the call.

NETWORK ISSUES

Although Versatility software is not dependent on a specific topology, generally our call centers are implemented over Ethernet, arguably the most popular (bus-type) network topology. Ethernet is a set of Local Area Network (LAN) standards specifying cable type, connection method, how data is transmitted, etc. Versatility has installed call centers using token-ring topologies as well. Routers can connect entirely different topologies in a Wide Area Network (WAN).

Protocols

The Versatility Series relies on the Transmission Control Protocol/Internet Protocol (better known as TCP/IP) as the transfer protocol of choice to transmit data across the network. This protocol is used by Windows 95 and NT (for 16-bit systems, TCP/IP is available as a network protocol for Windows 3.1 and 3.11).

User Datagram Protocol (UDP), a simple, unacknowledged protocol that is part of the IP protocol family, was relied upon heavily in earlier implementations of Versatility products. It is a broadcast protocol that does not guarantee data packet delivery. This method has a better chance of data delivery than a specific transmission protocol such as TCP/IP under catastrophic conditions, but there is no acknowledgement of receipt. This is still used today in internet-based applications.

Each computer requires an appropriate network interface card (NIC) to connect with the local network.

Thirty-two-bit Versatility implementations generally use Windows NT and Windows 95 operating systems. Sixteen-bit systems use Windows 3.1 or 3.11; see the *Versatility Product Compatibility Matrix* referenced in Section 5 of this document.

Multi-Tiered Networking Model

The concept of tiers addresses the interaction of splitting client/server applications into multiple functional units which are broken up between the client and servers. The functional units can be broken into three categories: The *presentation layer* or GUI, the *business rules* or business logic, and the *sharing of data.*

Essentially, a one-tiered model (stand-alone computer) is a system in which the application performs all data and workflow processing; it uses the operating system for low level I/O and display services.

A two-tier model involves more than one functional unit.; there are several possibililities. For example, a two-tier model might involve two applications running on the same desktop and sharing the OS. Or, it may involve execuction of the same application on two different machines using the same operating system, as when Versatility runs on an NT server and the client is an NT worksation. Alternatively, it might involve using a different operating system for some services – for example, having a database server process database requests and send the results to the client. Three-tier or multi-tier systems involve distributed computing to other systems for other services as well, with three or more functional units.

Versatility implementations combine different tiered models when using different services. For example, the client (agent's workstation running Versatility *Telesales/Teleservice* and/or *PowerGuide*) handles the presentation layer and some of the business rules layer, and the Versatility services and the relational database provides data for that application and those services.

CLIENT/SERVER AND VERSATILITY

Versatility software solutions take advantage of the following Client/Server features:

Scalability – When additional processing power is required by a call center, client/server technology provides for the addition or upgrading of servers easily as part of its intended architecture.

Fault Tolerance – In order to provide for security and ensure a call center's critical functions are not impaired at a single point of failure, client/server systems allow fault tolerance. This includes replicant servers that are hot-swappable in times of need. See *Guide to Creating Fault Tolerant Systems Configuration for Versatility Servers and Workstations* for a discussion of setting up Versatility solutions for maximum fault-tolerance.

Open Systems – Client/server architecture allows software development to allow middleware to handle communications between various hardware and software systems independent of the actual application program or service, allowing development to concentrate on their areas of expertise while providing functionality for diverse users of different systems using different protocols to use the same software.

Components of Client/Server

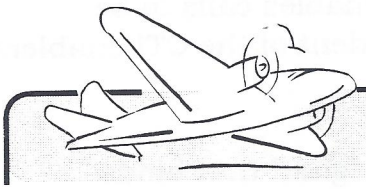
Client/server consists of the client, the server, and the middleware between the client and the server.

Client

For Versatility software, the agent's workstation is the main client, which administers the GUI, desktop workflow, and local application business rules. The client accesses distributed services – accessing the RDBMS, and using remote procedure calls (RPCs) to access other services as well. Note that the agent's workstation is not the only client; some servers can also be clients to other services as information is requested of them.

Server

The server runs software services such as those described throughout this document (list management, predictive dialing, etc.) to support *Telesales/Teleservice.* It provides those services to the agent application and other services.



ON THE HORIZON

In future versions of Versatility software, the business logic will be more segregated from our applications and services. At that time, the methods and properties within Versatility software will be easily exposed, and Versatility services will be able to support call centers without needing to access the base application.

Middleware

The software that ties together these components and data systems is referred to as "middleware." Middleware is comprised of platform-independent services that provide global, low-level enablement that operate *between the operating system and the application layer.* This is a broad category that can include:

- CTI server software enablers (such as Dialogic's *CT-Connect*) for telephony integration;
- Application Programming Interfaces (APIs), which are application- and operating system-specific software interfaces that pass information from one software component to another (most notably in Versatility, from each service to the CTI server);
- Distributed Computing Environment (DCE) software (discussed in the following section);
- Database connectivity enablers which reside on the database server and on each agent workstation to

translate generic SQL statements to the specific RDMBS being used;

- Versatility *OpenTel*, a software service that resides as a Dynamic Link Library (DLL) on each agent desktop which serves as an abstraction layer between the call center client and the CTI layer. *OpenTel* enables calls to be handled from the desktop independent of the CTI enablers required.

Middleware is often referred to as the "glue" that binds together the hardware systems that enable call centers to coordinate and manage the actual process of contacting or being contacted by existing clients or potential customers.

Again, middleware is a vague concept and arguments can be made both for and against the above examples as to whether they fit in this category. For example, since *OpenTel* does not communicate with the switch (PBX and/or ACD), some might not consider it middleware. Similarly, database connectivity enabler client software might be excluded by some from this category. The main issue is that some software exists – mostly third-party products – that enable communication between applications and services.

Distributed Computing Environment

The Versatility software suite is based on client/server architecture, in which some processes are distributed among two or more computers. Versatility software operates within a Distributed Computing Environment (DCE), a standard created by the Open Software Foundation and sometimes referred to as OSF DCE.

The Versatility client communicates with Versatility services through remote procedure calls (RPCs), which (along with registration, location, translation, time services, and memory/thread handling) are functions of DCE.

RPCs allow Versatility applications to call pre-built or pre-defined procedures from outside of that specific application's address space (whether on the same or a distributed server). In this way, an application can be distributed among multiple

computers in a manner that is highly transparent to the application-level code.

There are different implementations of the DCE standard. Versatility relies on Gradient Software's implementation, which is called *PC-DCE.*

In Versatility configurations using Dialogic's *CT-Connect* telephony enabler, Microsoft RPCs may also be used (specifically in reference to the Windows NT registry). However, Versatility solutions still require *PC-DCE,* since the other DCE functions mentioned above are also used.

At least one DCE cell is required to run any Versatility call center that includes the use of any services (*Campaign Plus* and/or *Predictive, CallBlending, Insight, IVR*). This DCE cell must be located on a server (although it may be on the same box as another service with modest resource requirements, particularly *Insight , IVR,* or the CTI server). In some cases, duplicate, "replicated" DCE cells are recommended. Cell servers must provide security (SEC), cell directory service (CDS) and RPC advertising services.

DCE is addressed in greater detail in a Versatility white paper by Ken Rush. An updated version of this document, entitled *DCE and Versatility,* is referenced in Section 5.

Versatility Automation Server

The *Versatility Automation Server (VAS)* enables developers who are building custom user-interface components in tools like *PowerGuide* and *Visual Basic* (VB) to access *Versatility Telesales/Teleservice*'s features and functionality.

Developers now have the power to build custom applications in easy-to-use tools without ever having to modify the base *Telesales/Teleservice* application.

VAS runs as an Automation Server specially designed to manipulate *Telesales/Teleservice* functions and data for use in *PowerGuide,* VB applications, or other development tools that support OLE. This server is only used with 32-bit Versatility implementations.

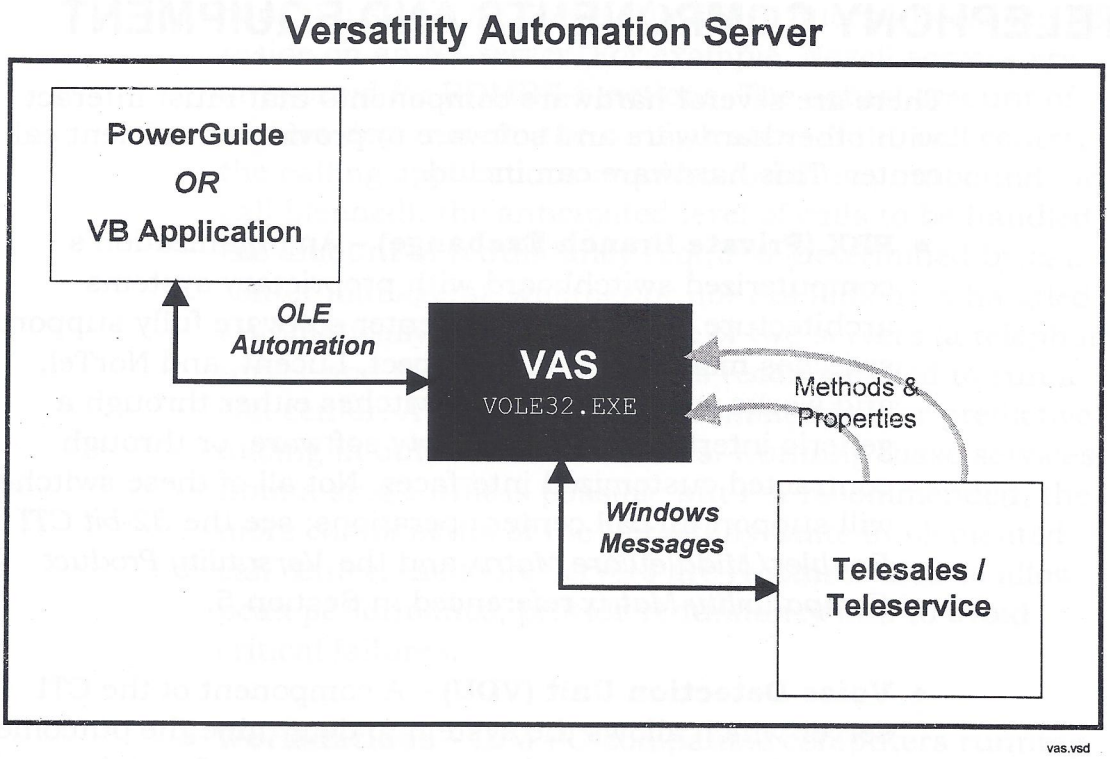
How VAS Works

Some call center managers elect not to use *Telesales/ Teleservice* as their primary agent interface for the call center. Whether they choose to use *PowerGuide* or build custom user-interface components in Visual Basic or other tools that communicate through OLE, they may still take advantage of the mature call center capabilities resident in Versatility software.

VAS serves as a desktop interface between *Telesales/Teleservice* and *PowerGuide* or the VB application, essentially translating between the two applications. It is a closed executable file (named VOLE32.EXE) that resides as a service on the agent desktop and is executed automatically when the *Telesales/Teleservice* application is started.

VAS exposes the methods and properties of *Telesales/Teleservice*, thereby providing interfaces into a majority of the features and functionality within the agent application, including all of the primary customer interaction functions below (list not inclusive):

- Customer (Info, History, Profile, Search, etc.)
- Transaction processing
- Product information and pricing
- Order, quote and Fulfillment generation
- Telephony
- Campaign Information
- Literature Fulfillment
- Callbacks



Each of the above functions is essentially a set of methods and properties. *VAS* maintains the integrity of the customer interaction framework and continues to enforce the workflow rules embedded in *Telesales/Teleservice*. Serving in its middleware role, *VAS* enables developers to access the functions of *Telesales/Teleservice* described above. **It is strongly cautioned** that a clear understanding of the *Telesales/Teleservice* business and workflow rules – especially as relates to the Versatility transaction – is a prerequisite for the successful development of call guides.

To reduce development time and limit unnecessary complexity, it is recommended from a development perspective to use *Telesales/Teleservice* for the functions inherent in it, and to limit the use of *PowerGuide*, VB, or other development tools to providing functionality *not inherent within Telesales/Teleservice*.

VAS also enables database calls via the SQL Engine component. The SQL Engine is useful for pulling information from the Versatility database in situations where the application is concerned about the number of connections to the database. Data retrieval times are often slower than using another connection to the database with ODBC.

TELEPHONY COMPONENTS AND EQUIPMENT

There are several hardware components that must interact with other hardware and software to provide an efficient call center. This hardware can include:

- **PBX (Private Branch Exchange)** – An organization's computerized switchboard with proprietary systems architecture. Versatility call center software fully supports switches manufactured by Aspect, Lucent, and NorTel, and works with many other switches either through a generic interface, via third-party software, or through contracted customized interfaces. Not all of these switches will support all call center operations; see the *32-bit CTI Enabler/Middleware Matrix* and the *Versatility Product Compatibility Matrix* referenced in Section 5.

- **Voice Detection Unit (VDU)** – A component of the CTI server which allows the system to determine the outcome of a call, e.g., connect, busy, no answer, or Special Information Tones (machine-detectable tones preceding network-provided announcements such as vacant, intercept, reorder and no circuit).

- **Interactive Voice Response (IVR) unit** – Equipment that prompts callers to input information about their accounts or about the nature of their call. These units can collect, store, and route data (customer information) and voice (the call) to various locations to provide for screen-pop, account handling, customer service, Fulfillment, etc. An IVR unit might also be used to direct inbound calls to particular agents based on customer-input information. For example, agents with particular skill sets (e.g., fluency in a particular language) may receive certain calls from responding to a message such as, "For English, press 1; para Espanol, toca 2." The IVR feeds such information into the ACD so that calls are correctly routed. Also sometimes known as a Voice Recognition Unit (VRU).

- **Server** – Any of several computers dedicated to serving a particular function or functions. The Versatility Suite primarily uses Windows NT servers; multiple services can

run on a single NT server. Database information need not reside on an NT server. For example, Novell servers are supported for RDMBS functions. The actual amount of servers depends on: the configuration of the call center, the calling applications required (inbound, outbound, or call blended), the anticipated level of calls to be handled, the amount of redundancy required (determined by call center managers), whether or not Fulfillment is handled automatically, etc. A minimum of two servers (a telephony server and a database server) is recommended to run a call center. A third server is recommended for predictive dialing in outbound call centers. Running these services from a single unit is possible but not recommended; the more components of the Versatility Suite implemented in a call center, the more servers are recommended to allow peak performance, provide redundancy and to avoid critical failures.

- **Workstations –** IBM PC-compatible computers running Windows software (95 or NT) from which Agents run their assigned campaigns or Supervisors monitor the call center. In 16-bit Versatility implementations, the OS may be Windows 3.1 or 3.11.

- **Automatic Call Distributors (ACDs) -** Equipment at a call center that directs incoming calls and provides reports on aspects of the call transaction. An ACD might direct calls to agents whose lines aren't busy, or to specific agents based on the Dialed Number Identification System (DNIS) number. The decision ability of ACD is limited; however, in combination with the information passed to it by IVR, a broad range of call direction is available. ACDs can also report on information such as the average length of each call or the longest amount of time a caller waited for an agent. The ACD can be part of the PBX, and may either be software-based or hardware *and* software-based.

HARDWARE COMPONENTS – GENERAL DISCUSSION

In order for a call center to fully utilize Computer-Telephony Integration (CTI) technology, it requires a variety of hardware. The bulk of these components fall into two categories: *telephony* (switches, routers, wiring, specialized equipment), and *computers* (agent and supervisor workstations, dedicated servers, and networking hardware to support that environment's topography (e.g., ethernet, token ring, etc.)).

Some hardware is essential to any call center, and other equipment is required only for certain configurations, depending on the call center's functions, workload, platform, and network protocols.

For example, any call center that meets our minimum definition requires a Private Branch Exchange (PBX) switch, which may include an Automatic Call Distributor (ACD), on the telephony side; a minimum of two servers (a database server and a CTI server); and agent workstations. Backup computers (to protect against mission-critical hardware failure) are highly recommended and should therefore also be considered a requirement.

Outbound call centers may require additional equipment; for example, on the telephony side, Voice Detection Units (VDUs) for predictive dialing, plus (on the computer side) at least one more server for any combination of predictive dialing, list management, and advanced reporting functions. Multiple services can run on one NT server, although ideal configurations have each service running on a single dedicated server. A matrix depicting the hardware/software dependencies for integrating various Versatility components is included later in this document (see the *Versatility Product Compatibility Matrix* referenced in Section 5).

Based on the call center's needs, these requirements can vary widely. When a call center's scope (set of *business rules*) changes, the corresponding hardware requirements often must change to meet that functionality. For example, for increased agent efficiency in an inbound environment, management may choose to install Interactive Voice Response (IVR) equipment to narrow the focus of the incoming call and route that call to an appropriate agent automatically.

Alternatively, computer requirements may grow to include expanded servers for larger databases; separate servers for predictive dialing, list management, and/or productivity reporting. A call center manager may decide to install a server to handle literature fulfillment, rather than handling that function manually. As a call center's needs change, managers should continuously evaluate existing configurations to ensure a comfortable level of redundancy to maintain data integrity or to support call center operations during crucial equipment failure. Call centers require robust hardware, which varies for different configurations. Future editions of this document will include a *Memory and Processor Considerations* section to address this.

SECTION 5 – BIBLIOGRAPHY/ REFERENCE MATERIAL

External Resources

Newton's Telecom Dictionary, 12th edition, Harry Newton, Flatiron Publishing (Feb 1997) (definition of SIT)

Client/Server Computing for Dummies, Doug Lowe (June 1995), IDG Books Worldwide Inc. (ISBN 1-56884-329-1)

Open Software Foundation DCE 1.1 Administration Course, Student Guide Version 2.0 (Volumes 1 and 2) (1995), Open Software Foundation, Inc.

The Essential Client/Server Survival Guide, Second Edition, Robert Orfali, Dan Harkey, Jeri Edwards (1996), John Wiley & Sons, Inc. (ISBN 0-471-15325-7)

Internal Resources

DCE and Versatility (White Paper), Ken Rush. Versatility, Inc. (1997)

Guide to Creating Fault Tolerant Systems Configuration for Versatility Servers and Workstations (White Paper), Ken Rush. Versatility, Inc. (1997)

Versatility Terminology Handbook, Versatility, Inc. – A comprehensive glossary of terminology that relates to call centers, telephony, and Versatility products.

Versatility Implementation Guide ***(Draft),*** Kathy Knill, Versatility, Inc. (1997) – Document that specifies process for implementations for "vanilla" (non-vertical-market-specialized) configurations of *Telesales/Teleservice.*

Special Thanks

Special thanks to the following individuals, without whom this document could not be produced. These are talented people who possess subject matter expertise, patience, and experience in programming, call centers, training, development, or business management.

Ken Rush, Scott Krall, Jim McDonald, Bill Ashton, Rick Day, John Kuzmicki, Kerry Murphy, Atul Uchil, Bruno Castegnaro, Tom Drewry, Mike Agha-Khan, Scott Klimek, Young Yu, Sumant Anand, Crystal Pilkins, Melanie Mann, Danita Thomas, Paula Hamm, Brian Carrier, Sudhir Rao, Keith Denson, Neil Ho, Bob St. Ledger, Steve Oviatt, Floret Ikome, Bob Sinesi, Dan Brigham, Tom Powers, Mitch Duvall, Kit Bigler, Kathi Knill, Kelly Waffle, Alice Carr, Steve Koski, Mike Rabatin, Amjad Ismail, and Jim Snee.

More people are added to the list every day; your efforts and time are certainly appreciated.