Lotus

SmartSuite 97 EDITION

The Best Suite For Today's Connected World — Powered by 1-2-3.

GETTING THE MOST OUT OF LOTUSSCRIPT IN SMARTSUITE 97

WINDOWS 95 and NT 4.0

COPYRIGHT

Contents

Introduction

This book introduces LotusScript® and describes how it extends the power of Lotus® SmartSuite® 97.

By the time you finish reading this book, you will have a broad understanding of the following points:

- LotusScript as a programming language
- How LotusScript and SmartSuite work together
- How applications created in LotusScript can improve productivity for you and for your workgroup
- The other parts of the SmartSuite application development environment: the Integrated Development Environment, Dialog Editor, and LotusObjects™
- The advanced programming features of LotusScript

This book also lists the documentation for LotusScript and explains how to get further help in developing LotusScript applications.

Read this book to learn about LotusScript

Whether you develop applications full time, direct the work of application developers, or write simple applications own your own, you'll find that LotusScript has a lot to offer.

If you are new to programming, you'll find that LotusScript is a good language to begin with. It offers considerable power in an easy-to-use environment. Many of the little details that you have to learn in other programming languages are handled automatically in LotusScript.

If you have experience writing macros in 1-2-3®, Ami Pro®, or Approach®, you can easily step up to LotusScript. In fact, your 1-2-3 and Approach macros run in SmartSuite 97; you do not need to convert them to LotusScript.

If you use IBM® VisualAge™ or Microsoft Visual Basic, or if you have written BASIC programs, you can easily transfer that programming knowledge to LotusScript. LotusScript is BASIC.

Obviously, the more experienced you are with programming languages, the more familiar the information in this book will seem; so some of you may need to slow down your reading only for the more technical sections of this book.

Regardless of the level of your programming experience, you should read this book if you are interested in adding LotusScript to the collection of tools that improve your productivity, or the productivity of your company, workgroup, or clients.

If you do not have the time to learn LotusScript or to develop LotusScript applications yourself, Lotus can assist you or someone in your department in locating appropriate technical support. Through the SmartSuite Business Partners program, for example, you can locate the names of companies in your area that specialize in the development of SmartSuite applications. Either way, be sure to hand off this document to the developer you work with. Refer to the last section of this book for more information on LotusScript documentation and support services.

SmartSuite 97: The tools you need to be productive every day

SmartSuite gives you the basic tools that you and your team need to be productive every day. You get the full-featured word processing of Word Pro® 97, the world-famous spreadsheet computing of Lotus 1-2-3, and the professional presentations of Freelance Graphics®. You can manage data in the relational databases of Approach 97 and manage daily tasks using Lotus Organizer®. Finally, there's ScreenCam®, an onscreen, show-and-tell teaching tool that lets you record and view movies of screen activity. All available at a click through SuiteStart and SmartCenter™.

Great products like those in SmartSuite 97 help you finish your work on time. Great programming tools help you get ahead of the work because you can develop applications that help you do your work faster, better, and smarter. All the tools you need to develop great applications are in your SmartSuite 97 package. And because you design these applications to your specifications, they will meet your exact needs.

What is LotusScript?

SmartSuite uses LotusScript, a version of BASIC that offers not only the standard capabilities of structured programming languages, but a powerful set of language extensions that enable object-oriented development within and across products as well. Using LotusScript, you can combine features of Lotus Notes® and the various SmartSuite products to customize, automate, and integrate these products as well as to create new applications.

That's the technical definition. Let's break it down a bit more.

LotusScript is a **full-featured programming language**. Unlike macro languages or batch processing languages, LotusScript offers all the programming power that you need to develop applications that are important to your business. You won't start programming in LotusScript for a few days only to discover that some important piece is missing from the language. It's all there.

LotusScript is **BASIC**. LotusScript, Visual Basic, and many other programming tools are based on the same, easy-to-learn language: BASIC. If you have worked with any of these other versions of BASIC, you will be on familiar ground during your first sessions with LotusScript.

LotusScript is **extensible**. Just as human languages evolve by adding new words and by incorporating new patterns of speech, programming languages evolve as well. With each new release of the LotusScript language, you get new capabilities and new tools for developing and maintaining the applications that you write. You can also add new objects and commands with the LotusScript Extension (LSX) Toolkit. LSX modules provide bridges between your current version of SmartSuite and late-breaking technologies. You won't outgrow LotusScript; it grows with you.

LotusScript is **object-oriented**. SmartSuite products like Freelance Graphics are composed of many objects — pages, diagrams, text, multimedia, and so on. To develop applications that manipulate these objects, you need a programming language that knows the structure of objects — classes, methods, properties, events, inheritance, and containment. With LotusScript, you can work with all the objects in SmartSuite products, OLE objects in the Windows environment, and new classes of objects that you define as you go.

LotusScript is a **cross-product language.** SmartSuite products all speak the same language: LotusScript. SmartSuite objects share the same design. For example, from inside Word Pro, not only can you give instructions to Word Pro, but you can also use the language of 1-2-3 objects to give instructions to 1-2-3. Later, you'll learn that you can also use OLE Automation and LotusScript to program objects embedded from other Windows products.

Want to see it?

So what does LotusScript look like? In LotusScript, you build applications from the bottom up, statement by statement. Collections of statements form a script. Collections of scripts in a document form a LotusScript application.

What follows is just a sample. It’s a simple script containing six statements and five comment lines (those lines beginning with a single quotation mark). This script is contained in a structure called a sub (short for “subroutine”). Other types of scripts perform declarations, set options, define classes, and define functions. This sub tells 1-2-3 to do several things: create a document, fill a range in that document with numbers, sum columns of numbers in that range, and change the background color of the cells containing those sums.

```
Sub FillNewSheet
   ’ Comment: Create a new document.
   CurrentApplication.NewDocument
   ’ Select a range of cells and fill it with data.
   [A:A1..A:D18].Select
   Selection.RangeFill "0","1","32767",$Number,
   ’ Select cells below the range to hold totals.
   [A:A18..A:D18].Select
   ’ Use SmartSum to create the toals.
   Selection.SmartSum
   ’ Change the background color and fill for the totals.
   Selection.Background.Color.ColorName = "leaf green"
End Sub
```

It doesn’t take a lot of LotusScript code to accomplish powerful tasks.

Want to try it?

Do the following to create a very basic script.

1. Start any one of the SmartSuite applications that support LotusScript: 1-2-3, Approach, Freelance Graphics, or Word Pro.
2. If you start 1-2-3, Freelance Graphics, or Word Pro, choose Edit - Script - Show Script Editor. If you start Approach, choose Edit - Show Script Editor.

3. Choose Create - Sub.

   The Create Sub dialog box appears.

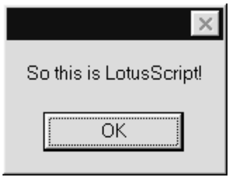
4. In the "Name" box, type **samplesub** and click OK.

   The blinking I-beam in the Script Editor indicates that the editor is ready for you to type your first LotusScript statement.

5. Type the following statement and press **ENTER**:

   ```
   MsgBox "So this is LotusScript!"
   ```

6. Choose Script - Run Current Sub to see the script execute and display the following message box.

   So this is LotusScript!

7. To close the message box, click OK.

This one statement does a lot — displays the message box, displays your message in the box, adds an OK button, and defines the behavior of the message box when you click OK.

LotusScript means business

So, LotusScript is easy to use and powerful, but will it help you solve practical, day-to-day business problems? Will it support the kinds of tasks that make a difference?

To answer this, consider the tasks that you perform. Some of them are simple tasks that you do over and over. Others are probably more complicated and may require you to use more than one product. Yet others are critical tasks, requiring you to gather information from several sources and make judgments based on that information. Not all this information is necessarily on your PC or stored in a SmartSuite product.

If these kinds of tasks sound familiar, you’re going to like the way LotusScript and SmartSuite work together to make your work more productive. Let’s take a closer look at the ways you can use SmartSuite 97 and LotusScript to accomplish these business tasks: automating frequent tasks, customizing applications, and integrating information from more than one product.

Automating frequent tasks

There are some tasks you probably find yourself doing over and over again. When you think about it, there are probably dozens of repetitive tasks you do: tasks that involve a series of steps or require integrating files from one SmartSuite product into another. Wouldn't it be great if you could get jobs like these done in a single keystroke?

- Instantly incorporate the latest currency exchange rates into your expense reports
- Automatically import up-to-the-minute customer information into your sales presentations
- Get monthly data from your corporate files and quickly turn it into a graph

That’s where LotusScript comes in. LotusScript lets you create and use that Update Exchange Rates button — the one that accesses Web site currency data, copies the latest currency exchange rates, and pastes the data into your expense report spreadsheet. You can even have the final document formatted. All those tedious steps can be reduced to a single click of a button that starts a LotusScript application.

Best of all, you don’t have to be a programmer to automate simple tasks. In 1-2-3, Approach, and Word Pro you can perform a task while the product automatically records your mouse clicks and keystrokes as working LotusScript code. To perform that task again, simply run the recorded script.

Customizing SmartSuite products

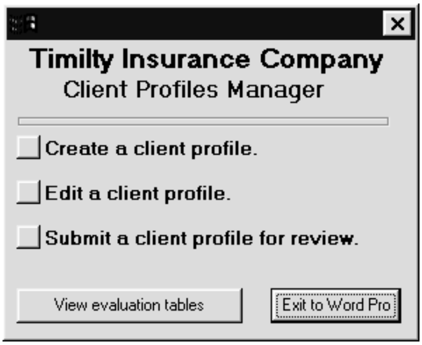
When you install SmartSuite, all the menus, SmartIcons®, display options, and commands appear in a default configuration. Although these defaults work for most people and for most tasks, you may find that the way that your workgroup uses a product may differ from the way the defaults are set up. You may want to modify the menus or hide SmartIcons that people do not use or create a custom dialog box assisting people through a difficult task. This is all possible with LotusScript.

All the menus, SmartIcons, commands, and display options in your SmartSuite products are programmable with LotusScript. You can write scripts to modify the defaults or to create an alternate interface for the tasks that your group performs most frequently.

An example

Suppose that your underwriting department uses Word Pro to build profiles of potential clients. To evaluate clients consistently, the profiles must be developed with accurate data and in a consistent format. You could write a five-page memo explaining how to create and manage these profiles, but there's a better solution with LotusScript.

You can use the Lotus Dialog Editor to create a custom dialog box that presents people in your group with the most common tasks for managing client profiles. People in your group can launch this dialog box every time they start Word Pro or only when they need it. Adding the name of this dialog box to a Word Pro menu is easy. If the document containing the dialog box is stored on a department server, then you can make changes to that master copy of the dialog box and everyone in your department will see the changes automatically. When they need to create, edit, or submit a profile, they click a button in the dialog box and Word Pro handles the mundane tasks of loading the appropriate SmartMaster™ and retrieving the appropriate default values for the SmartMaster. Here's how it might look.



Timilty Insurance Company

Client Profiles Manager

□ Create a client profile.

□ Edit a client profile.

□ Submit a client profile for review.

You might even use LotusScript to launch other SmartSuite products from this custom dialog box. For example, the button labeled "View evaluation tables" could start 1-2-3 and open a workbook containing tables for evaluating clients.

Customizing SmartSuite products so they support your work helps you manage change. You can update a dozen memos each time your business encounters a new requirement or you can invest in LotusScript to manage new requirements centrally.

Integrating information

If your organization stores important information in several products or at multiple sites, you'll like the way LotusScript helps you integrate that information and make it work for you. Gathering the information and getting it into shape for a report can be time-consuming work. And sometimes not all of that work is really necessary. If the tools that contain the information are not designed to work together, you can spend more time reformatting the information than analyzing it.

SmartSuite is already great at integrating information. SmartSuite products are designed to share text, graphics, multimedia, database tables, and spreadsheet ranges. LotusScript provides the glue to assemble and update these pieces of information.

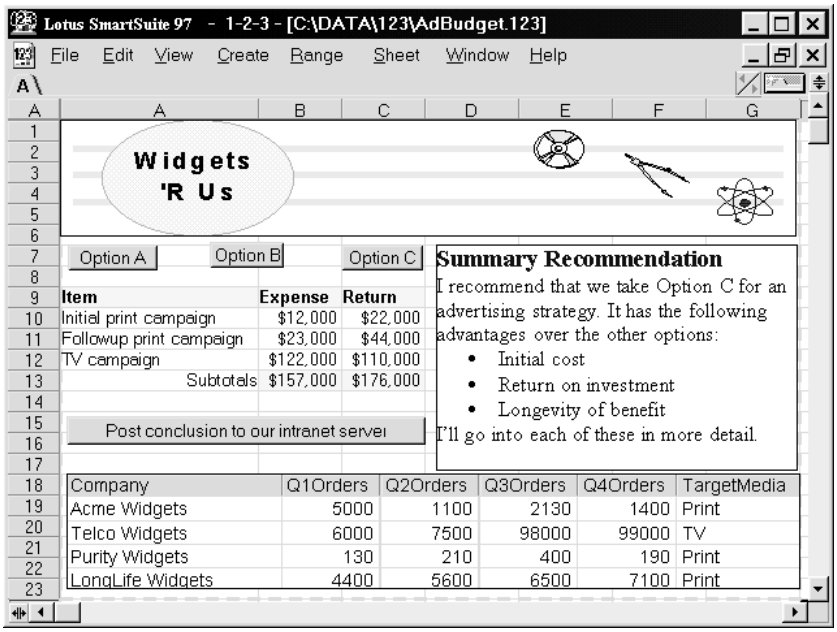
An example

Suppose that when your boss develops an ad campaign, she always asks you to prepare the same one-page report that extracts raw data from the mainframe, analyzes best and worst scenarios about advertising strategies, offers a recommendation, and lets her play with the numbers a bit in a live document. Sound bad? Not really.

Here's how SmartSuite and LotusScript could work together to integrate information from multiple sources and assemble all that information in a compound document.

1. Use Approach to query the raw data on the mainframe and build a local database of relevant information.
2. Use Freelance Graphics to customize your company's logo.
3. Use Word Pro to write your final recommendations and to format them for impact.
4. Use 1-2-3 to pull it all together:
   - Embed the logo from Freelance Graphics at the top of your report.
   - Embed the Approach table containing the information from the mainframe.
   - Embed the Word Pro document next to your summary range.
   - Create a 1-2-3 range that summarizes the information in the Approach table.
   - Create a command button in 1-2-3 for each scenario you want to put in front of the boss: Option A, Option B, and Option C. You can attach LotusScript code to each button so it modifies the data to suit your scenarios. Want to see the numbers behind Option A? . . . click the Option A button.
   - Create another command button in 1-2-3 that saves the final contents of the sheet to your intranet server.

Here’s how it looks when you send it to the boss:


Lotus SmartSuite 97 - 1-2-3 - [C:\DATA\123\AdBudget.123]
File Edit View Create Range Sheet Window Help
A
Widgets 'R Us
Option A
Option B
Option C
Post conclusion to our intranet server

Item	Expense	Return
Initial print campaign	$12,000	$22,000
Followup print campaign	$23,000	$44,000
TV campaign	$122,000	$110,000
Subtotals	$157,000	$176,000

Summary Recommendation

I recommend that we take Option C for an advertising strategy. It has the following advantages over the other options:

- Initial cost
- Return on investment
- Longevity of benefit

I’ll go into each of these in more detail.

Company	Q1Orders	Q2Orders	Q3Orders	Q4Orders	TargetMedia
Acme Widgets	5000	1100	2130	1400	Print
Telco Widgets	6000	7500	98000	99000	TV
Purity Widgets	130	210	400	190	Print
LongLife Widgets	4400	5600	6500	7100	Print

SmartSuite 97 provides the building blocks. LotusScript glues them together and makes them interactive. You provide the expertise. And simply by changing the sources of your information with a little tinkering in LotusScript, you can produce this report over and over again.

Developing applications in LotusScript: A technical overview

You’ve now seen some of the exciting opportunities for improving your productivity in SmartSuite by using LotusScript. Now how do you make them happen? This section provides a more technical overview of the development process and development tools. If you find that this gets too detailed, you might skip to the final section on planning your next steps with LotusScript.

An overview of the LotusScript development process

Whether you're writing a simple script to automate a task or you're developing a large application, the process is essentially the same. Here are the basic steps you follow to develop a LotusScript application:

1. Create some objects like text, graphics, or tables in a SmartSuite product. These are called LotusObjects.
2. Write some LotusScript code in the Integrated Development Environment (IDE) to program the behavior of these objects. The code can be stored in one script in your document or distributed across multiple scripts and documents.
3. Optionally, add some code that takes advantage of the advanced programming features in LotusScript.
4. Run and debug the scripts in your document.
5. Save the document containing your scripts.
6. Distribute the document containing the scripts to others in your workgroup.

The process is straightforward. Your SmartSuite products provide the objects that you need to build great applications. LotusScript provides the programming language, the programming tools, and user assistance to make these objects work for you.

The same programming language. The same programming tools. Similar objects across products. SmartSuite and LotusScript are designed to get you programming on the first day.

Let's take a more detailed look at each of the elements in this process.

The LotusScript Integrated Development Environment and its Dialog Editor

All the tools you need to write LotusScript applications are built into your SmartSuite applications. The Integrated Development Environment (IDE) includes the Script Editor, Script Debugger, Class Browser, and Dialog Editor. Plus, you get great online Help, documentation, and sample applications to get you going.

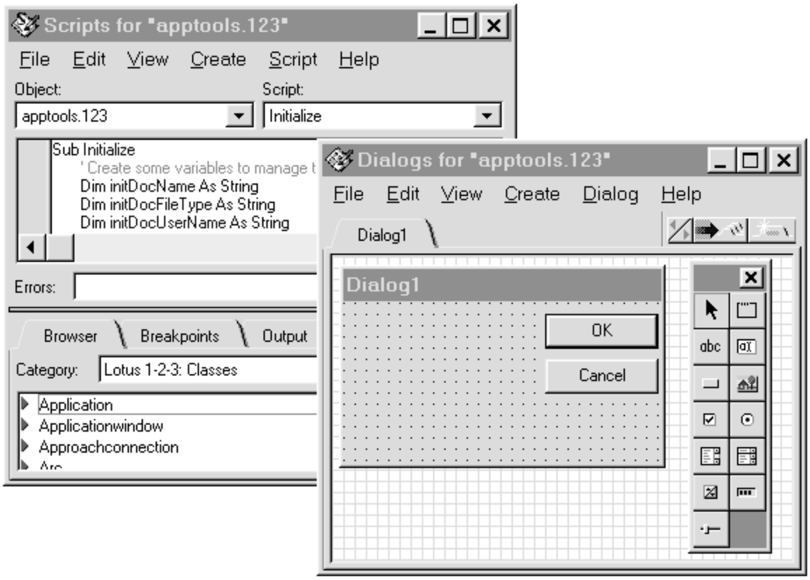
The LotusScript IDE offers all the specialized programming tools that you would expect in a professional programming environment. If you have used the development environments of other scripting languages, you'll be on familiar ground with the LotusScript IDE. There are also some ways that the LotusScript IDE is different from the environments for Visual Basic or other scripting languages:

- The IDE is not a stand-alone application. Instead, the IDE is incorporated into each SmartSuite product that supports LotusScript, so you get the full complement of development tools where you need them.
- Script applications that you write in LotusScript are stored in SmartSuite product documents. You do not have to create separate files for your scripts, dialog boxes, or data files.
- The IDE does not use a separate window for each tool. IDE tools are conveniently integrated into one IDE window. The Dialog Editor is the only tool that uses a separate window.

To display the IDE window, choose Edit - Show Script Editor (in Approach) or Edit - Scripts & Macros - Show Script Editor (in 1-2-3, Freelance Graphics, and Word Pro).

To display the Dialog Editor in 1-2-3, Freelance Graphics, or Word Pro, choose Edit - Scripts & Macros - Show Dialog Editor.

Here's what the IDE window and the Dialog Editor look like in a SmartSuite product:



Object: apptools.123

Script: Initialize

```
Sub Initialize
    ' Create some variables to manage t
    Dim initDocName As String
    Dim initDocFileType As String
    Dim initDocUserName As String
```

Category: Lotus 1-2-3: Classes

- Application
- Applicationwindow
- Approachconnection
- Arc

You don't have to rearrange windows constantly to get your work done in the IDE. All your programming tools are readily available in these two windows:

Programming tool	*Programming tasks*	*Window*
Script Editor	Create scripts, add code to scripts, set breakpoints, run scripts, import and export text files containing script code.	IDE
Script Debugger	Locate script errors, monitor calls between scripts, monitor variables in executing scripts, step through scripts as they execute.	IDE
Class Browser	Review the structure of product objects and syntax for their properties, methods, and events.	IDE
Object selector	Select product objects that you want to program.	IDE
Script selector	Select scripts for a selected product object.	IDE
Dialog Editor	Create dialog boxes, add Lotus controls or third-party ActiveX controls to dialog boxes. (Not available in Approach 97.)	Dialog Editor

To make using LotusScript and the IDE as easy as possible, SmartSuite includes online Help and online documentation available from all SmartSuite applications.

To access Help for LotusScript, choose Help Topics from the Help menu to display the main Help contents. Then click the book called "LotusScript."

Within the Script Editor, you can get context-sensitive Help on any LotusScript statement or any LotusObject. Simply select the statement or object and press **F1**.

LotusObjects: The SmartSuite application programming interface

SmartSuite products can be easily automated, customized, and integrated because not only do they have compatible user interfaces and functionality, but they also share a common application programming interface (API).

The SmartSuite API is composed of LotusObjects, which are blocks of code that describe a component of a SmartSuite product. LotusObjects are specific to Lotus products. Each Lotus product that uses LotusScript has its own set of LotusObjects that can be programmed with LotusScript. Through LotusScript and the IDE, these objects are at your command; you can bring them into any application that you write.

All this object talk may sound mysterious, but it’s not. When you use the mouse and keyboard to accomplish tasks in a product, you are using the product’s user interface to control objects such as spreadsheet cells or query tables or paragraph styles. To control these objects from a LotusScript application, you need to talk to the objects in a language that they understand (LotusScript), using a vocabulary they understand. The vocabulary of Lotus Objects comprises properties, methods, and events.

Properties

Objects have characteristics called properties that allow you to control the way an object looks or acts. Properties have predefined values that you can change or set to control the behavior or appearance of an object. For example, in a script you can set the Visible property of an Approach FieldBox object to TRUE, causing the field box to be displayed. Similarly, when you name the FieldBox “IncomeSummary,” you are changing its Name property.

Methods

Objects also have methods, which are actions performed on or by the objects. Think of methods as the behaviors that an object understands. The actions change the object or give you control over certain aspects of the object’s behavior. For example, the FieldBox object has a Refresh method, which refreshes the value in a field box.

Events

Most objects respond to one or more events. When an object in your document receives an event, for example, a mouse click, it looks for a script associated with that event and executes it. For example, each FieldBox object in an Approach form can have a script associated with the click event. To make FieldBox1 display a message box when it is clicked, add a message box statement to the Click script for FieldBox1. By writing scripts for objects and their events, you determine the overall behavior of your application.

Although objects differ in individual SmartSuite applications, they share a common design. This allows you to manipulate one SmartSuite application’s objects with another SmartSuite application.

Advanced programming features

The LotusScript language is as flexible as it is powerful. It is designed to take advantage of new technologies and operating systems as they appear.

Here’s an overview of some of the advanced programming features in LotusScript that you may want to explore.

Integration with Notes

SmartSuite and Notes™ share the same LotusScript language. Whether you are a Notes power user or an occasional discussion database fan, you may want to explore the integration between SmartSuite products and Notes. You can use LotusScript to exchange data between SmartSuite products and Notes databases. Even better, you can embed SmartSuite objects in Notes and use LotusScript to develop workgroup applications. This provides a way to access and manipulate Notes objects, allowing you to take advantage of the power of Notes in your applications: its unique replication model, security, integrated messaging, and groupware infrastructure. This integration is enabled by the Notes LSX included in the products that support LotusScript.

OLE Automation

OLE Automation is the protocol that LotusScript uses to communicate between products. OLE provides a consistent and open standard for defining what an object is and how objects interact. Using OLE, you can seamlessly link data between two OLE2-compliant applications. For example, you can embed a Microsoft Project Gantt chart in your Approach project scheduling database and use LotusScript to manage the way Project and Approach share information.

Calls to dynamic-link libraries

If you have used other programming languages such as C++ or Visual Basic, you've probably developed some useful functions and then compiled them into dynamic-link libraries (DLLs) for general use. It doesn't matter what language you used to write the functions, LotusScript lets you call them in their DLL files and use the results of those functions in your script application. For example, your company may have some internal formulas for calculating an overseas loan. You can send information to that DLL function from LotusScript and get the results back to your script. Any of your script applications can call any of these DLLs.

Calls to the Windows operating system

The Windows 32 application programming interface consists of thousands of C functions that manage your desktop and the services available to you. You can use any of these Windows functions in your script application by writing a LotusScript function that calls the Windows function. For example, you can use the SetWindowText function, available in the Windows User library, to change the title bar text of the window from which you are running LotusScript.

New LotusScript classes in LSX modules

Lotus offers extensions to the LotusScript programming environment called LSX modules. Like DLL files, LSX modules contain code that you can call from your scripts. LSX modules also contain new LotusScript classes that you can add to your SmartSuite product the way you add a new sound card or hard disk to your PC. You can extend the number of objects in your SmartSuite product anytime you want.

Besides the Notes LSX in your SmartSuite package, other LSXs are being developed by Lotus and third-party vendors. LSXs are built by C++ programmers using the LSX Toolkit, which you can obtain from the Lotus World Wide Web site. These additional LSXs add new functionality that can be programmed from within the SmartSuite or the Notes IDE.

Internet integration

SmartSuite and LotusScript are ready for the Internet today. SmartSuite products support features that read information from the Internet, write data to the Internet, and manage the conversation between your PC and the Web. These Internet features are programmable through LotusScript, so you can develop applications for SmartSuite that exchange entire files or smaller chunks of information with Web servers.

LotusScript, macros, and other scripting languages

SmartSuite and LotusScript let you capitalize on what you already have — macros, Visual Basic scripts — while letting you take advantage of important new technologies: object-oriented programming, cross-product versions of BASIC, OLE Automation, ActiveX, and Web programming. Whether your company wants to preserve its investment in macros or Visual Basic scripts, SmartSuite and LotusScript are designed to work in open environments.

LotusScript and macros

It’s true, macros do a lot to help you automate tasks. In some SmartSuite applications, you can also associate macros with events such as clicking a button or icon, choosing a menu option, or opening a file. Macro support, however, differs from product to product across SmartSuite:

- In Word Pro, LotusScript replaces the macro language. Word Pro will run many of your existing Ami Pro macros, but you will not be able to write new macros.

- In 1-2-3 and Approach, keystroke macros and macro commands continue to be fully supported, so you can run existing macros, create new ones, and call macros from more sophisticated LotusScript programs.

If you're comfortable writing macros and they're giving you all the functionality you need, you can save LotusScript for another day. But if you want to do more with your SmartSuite applications, consider some of the advantages that LotusScript offers over macros:

- Improved maintenance
- Closer integration with SmartSuite product features
- Object-oriented support
- Cross-application support
- More sophisticated programming features

You don't have to abandon your investment in macros that do the job, but you'll discover that the more quickly you invest in LotusScript, the more quickly you'll see the benefits of a more powerful programming environment.

LotusScript, Visual Basic, and other OLE scripting languages

If you develop applications in Visual Basic, you've got a jump start on programming in LotusScript. LotusScript and Visual Basic share the same programming language, so you can leverage your knowledge of Visual Basic programming and syntax to use LotusScript.

OK, you say. If LotusScript shares the same language as Visual Basic, why not just continue to use Visual Basic?

Good point. If you want to add some SmartSuite functionality to a Visual Basic application, it's probably easier to start from your existing Visual Basic application. Since 1-2-3, Approach, Freelance Graphics, and Word Pro support OLE Automation, you can easily call these applications from within your Visual Basic code. In fact, you can use any OLE scripting language, such as C++ or Delphi, to include a component from one of the SmartSuite applications in a non-Lotus Windows application.

Generally, applications written in Visual Basic are larger than those written in LotusScript. Still, there are times when it's simply more convenient to use Visual Basic or some other OLE scripting tool. Lotus understands that and fully supports the use of Visual Basic and other OLE servers in SmartSuite 97.

Of course, if you are working mostly within SmartSuite 97, customizing or combining SmartSuite functionality, it makes more sense to use LotusScript. For one thing, the IDE is already built into each SmartSuite application that supports LotusScript. That means you can easily control the components within one SmartSuite application from any of the others.

Also, by using existing LotusScript extension files, you can view and use Notes components from within the IDE — something Visual Basic and other OLE scripting tools simply can't do. Only LotusScript gives you this level of SmartSuite and Notes integration.

Getting more information on LotusScript

There's a lot of good information about LotusScript programming in your SmartSuite package. It will clarify what LotusScript can do specifically to support your requirements.

- Online Help. You can scan Help to find information about LotusScript functions and statements, the IDE, and LotusObjects from the following Lotus products: 1-2-3, Approach, Freelance Graphics, and Word Pro.

  Note When you install each SmartSuite product, you must choose the Custom option and select each LotusScript Help file to install it on your computer.
- The *LotusScript Language Reference*, a comprehensive summary of the LotusScript language, presented in A-Z format.
- The *LotusScript Programmer's Guide*, a general introduction to LotusScript that describes its basic building blocks and how to put them together to create applications.
- *Developing SmartSuite Applications Using LotusScript*, an in-depth explanation of SmartSuite scripting including product-specific information and example scripts.
- LotusScript Web page, the most current source of publications, code examples, and white papers about LotusScript programming in SmartSuite (http://www.lotus.com/smartsuite/sslotusscript).

To obtain printed copies of the *LotusScript Language Reference*, *LotusScript Programmer's Guide*, and *Developing SmartSuite Applications Using LotusScript*, fill in the coupon for the Lotus SmartSuite Application Developer's Documentation Set, included in this pamphlet. Send the coupon and the specified shipping and handling fee to the nearest distributor listed on the back cover of this booklet.

Writing scripts: Should you do it yourself or seek help?

Now that you have an idea of what LotusScript can do, you need to decide who's going to be doing the programming.

If you'd like all the benefits of LotusScript, but you don't want to write the scripts yourself, here are some suggestions on finding qualified people to help write your LotusScript programs.

You can often get help from your internal MIS department. Anyone with Visual Basic or OLE skills can build LotusScript solutions. Lotus Notes developers are also a good source of scripting help, because LotusScript is also part of the Notes application development environment.

You can also get expert help by contacting any SmartSuite Ready Lotus Business Partner. For a list of Lotus Business Partners, visit the Lotus home page at http://www.lotus.com/ssbp. The Lotus Web page also includes details on how you and your coworkers can get LotusScript training.

If you work in a small company and don't have access to internal MIS help, contact your local computer society. They most likely can supply you with a list of application developers who have LotusScript or Visual Basic experience.

Fulfillment Information

Refer to the following table when ordering the Application Developer's Documentation Set for SmartSuite 97.

- Identify the fulfillment center/centre for your country.
- Use the appropriate shipping and handling charge for your country when ordering.

Note: The following charges include VAT (Value Added Tax) where applicable.

Country	*Charge*	*Mailing Address*
Australia	AU$35	Lotus Development Pty. Ltd. Customer Service Department Level 12, 321 Kent Street Sydney NSW 2000 AUSTRALIA
Canada	CD$15	Lotus Development Corporation SmartSuite Documentation P.O. Box 679 Scarborough Ontario M1K 5C5 CANADA
Belgium France Italy Netherlands Portugal South Africa Spain	BFr 900 Fr 150 ItL 45000 NGD 50 Esc 4500 US$30 Pts 3500	Lotus Assistance SARL Parc Club Ariane Bat. Neptune 5 Blvd. des Chenes, BP 219 78051 St. Quentin en Yvelines Cedex FRANCE
New Zealand	NZ$40	Lotus Development New Zealand Ltd. Customer Service Department Level 20, ASB Bank Centre Cnr. Albert & Wellesley Sts. Auckland NEW ZEALAND
Austria Germany Switzerland	ÖS 300 DM 45 SFr 35	Lotus Development Gmbh Baierbrunnerstrasse 35 Postfach 70 12 20 81379 Muenchen GERMANY
United States	US$10	Lotus Development Corporation SmartSuite Documentation P.O. Box 25367 Rochester NY 14625-0367 USA
Denmark Finland Ireland Norway Sweden United Kingdom	Dkr 175 Fmk 130 IR£15 Nkr 190 Skr 200 UK£15	Lotus Development European Corporation Lotus Park The Causeway Staines Middlesex TW18 9AG ENGLAND
Other Countries in Europe, the Middle East, or Africa	US$30	Lotus Assistance SARL Parc Club Ariane Bat. Neptune 5 Blvd. des Chenes, BP 219 78051 St. Quentin en Yvelines Cedex FRANCE

SmartSuite 97 Application Developer's Documentation Set Order Form

The following books about LotusScript are available on the CD-ROM version of your SmartSuite package as Online Books, on the Lotus World Wide Web site (http://www.lotus.com/smartsuite/sslotusscript), and in printed form in the SmartSuite Application Developer's Documentation Set. These manuals are available only in English.

- ***Developing SmartSuite Applications Using LotusScript*** is a comprehensive introduction to developing applications for SmartSuite 97. It offers chapters on key programming concepts, using LotusScript programming tools, programming individual SmartSuite products, developing cross-product scripts, and integrating SmartSuite scripts with Lotus Notes.
- ***LotusScript Language Reference*** provides a comprehensive summary of conventions and basic commands in the LotusScript language. *LotusScript Language Reference* provides the foundation for programming any product that supports the LotusScript programming language.
- **The *LotusScript Programmer's Guide*** describes the basic building blocks for creating LotusScript applications and provides many working examples.

If you want to order these printed books in the SmartSuite 97 Application Developer's Documentation Set, complete this form and send it to the appropriate office (see addresses on the back cover of this booklet). US customers can pay by check (payable to *Lotus Development Corporation* and drawn on a US bank); others must pay by credit card.

- -

Please send me a copy of the SmartSuite 97 Application Developer's Documentation Set (Part # XA-086305). I understand that the only additional charge for this documentation is the shipping and handling fee applicable to my country, provided that my request is postmarked no later than December 31, 1997. Please allow 4 - 6 weeks for delivery.

Name ____

Company ____

Address (no P.O. boxes please) ____________________________________

____

____

City ________________ State/Province _______ Zip/Postal Code __________

Country __________________ VAT# (if applicable) ______________________

Phone (____) _____ - _________ ext. _________

FAX (____) _____ - _________

Can we contact you electronically? Yes ___ No ___

If yes, please provide your email address:

____

Shipping and handling charge (see the back cover of this booklet): _____________

Payment method:

□ Discover □ Visa □ Mastercard/Access/Eurocard □ Check (U.S. only)

Card Number _____________________________ Exp Date ________ / _______

Signature ____

XA-086305