

Table Of Contents

SCRIPT ADMINISTRATION OVERVIEW	1
Overview	1
Call Flow Scripts.....	2
OPTIONS.....	4
Menu Bar.....	4
File Close	4
View Always on Top	4
Preferences Script Editor... ..	4
MANAGING SCRIPTS	5
Create and Edit Scripts.....	5
Import and Export Scripts.....	5
Route Calls to Scripts	6

Script Administration Overview

Overview

NexTalk has always been highly flexible and adaptable to customer's needs.

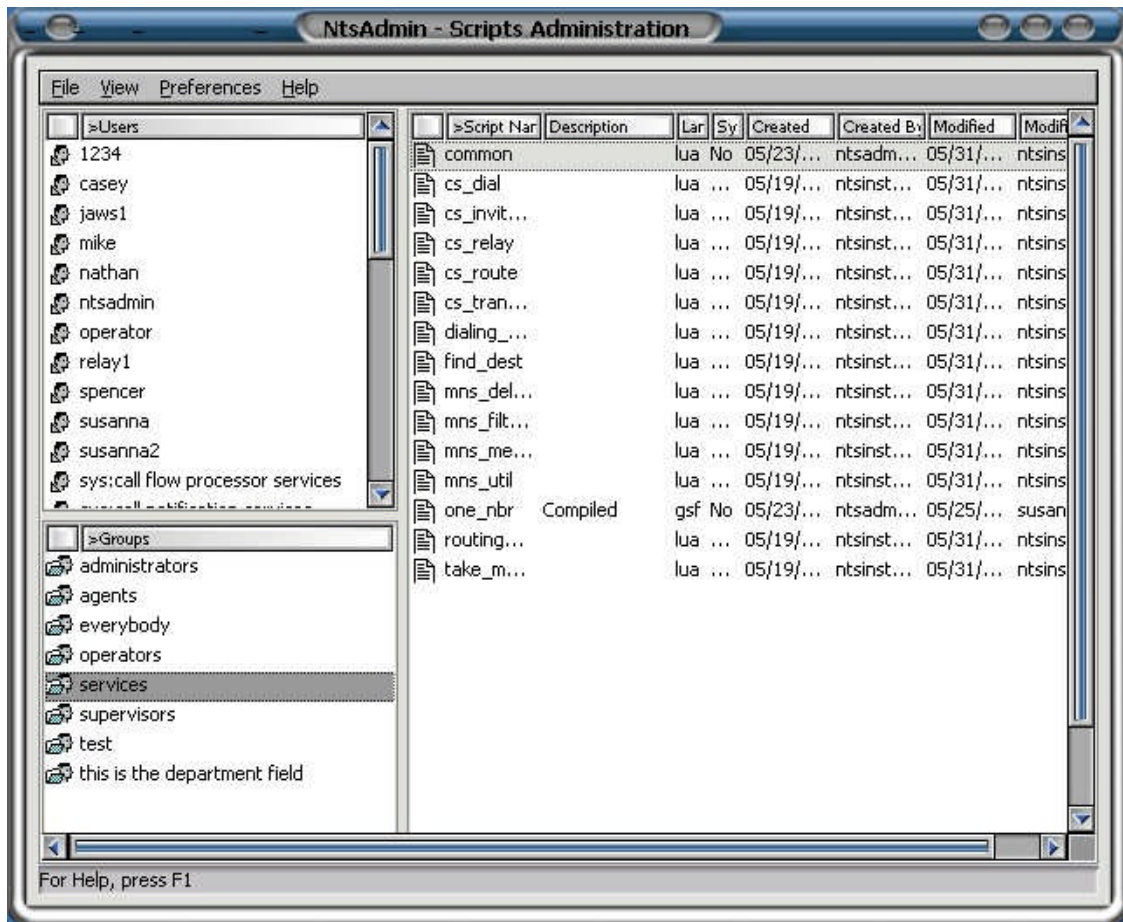
This flexibility is largely achieved using “**scripts.**” An NexTalk script is a series of instructions that the **CFP** module executes when handling calls. There are two types of scripts: **system scripts** and **user scripts**.

NexTalk uses system scripts to regulate much of its internal operation. For the most part, the operation of these scripts is hidden from the user. System scripts are written in a script language called “Lua” (see www.lua.org). The lua scripting language is flexible and powerful. There are very few if any of the system scripts that will need to be modified beyond the defaults we have provided. **Do not edit these scripts unless you are an expert in NexTalk or Lua, or our technicians are walking you through the process.**

A user script is any script written by an NexTalk user. Most user scripts are written as a “**Gsf**” file, created using an easy to use drag and drop **GUI**. User scripts can easily be edited and customized to the users needs.

NexTalk scripts can be opened, *imported, and exported* from the Scripts Administration module.

Script Admin



On the left side of the Scripts Administration window there are two list boxes; one of domain users and one of domain groups. Scripts are “**owned**” by users or groups. To view the scripts owned by a particular user/group, select the user/group name. The scripts owned by that user/group will appear in the main list box to the right. A description of the script, the language it is written in, the date it was created and modified, and who created or modified it, will be shown next to the script name. To view system scripts, select the “**Services**” group from the list.

Call Flow Scripts

A **call flow script** is a special type of system or user script. After a call is answered, a call flow script controls the sequence of events that follow.

Equivalent to IVR systems that provide automated call handling for voice callers, call flow scripts provide automated call handling for text-based callers (ITR). An IVR system may offer call transfer options with a recording that says “Press one for sales, two for technical support, or enter an extension number for a direct call transfer.” An NexTalk Call Flow Script can provide the same call transfer options for customers or clients calling via a TTY or web browser. The call flow script

requests and recognizes the text responses of a caller and then routes the call accordingly.

Incoming TTY calls from deaf callers are commonly routed to a call flow script. Routing is done using the Routing Table found in the Telephony Services module. Any incoming telephony call can be directed to any script by this means. By routing incoming calls to a script, callers will automatically be given transfer options or other important information before speaking with someone. Please see the Telephony Services manual for more information on call routing.

Call flow scripts can be very simple or very complex, depending on the needs of your organization. Two specific applications of call flow scripts are the One-Nbr™ service and Placeholder Calls (see TID's: Placeholder Call and One-Nbr™ service).

Options

Menu Bar

File | Close

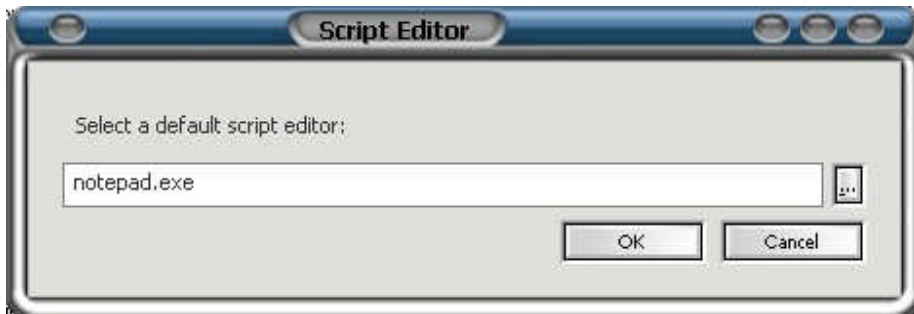
This option closes the Scripts Administration module.

View | Always on Top

This option keeps the Scripts Administration window in front of all other open windows.

Preferences | Script Editor...

The Script Editor option allows you to specify which program is used to open Lua script files for editing.



By default, the script editing program is set to Notepad. To browse for a different program, choose the "...” button to the right of the program text box. Highlight the appropriate program and choose Open. The file path and name of the new program will appear in the program text box. Choose Ok to save the change.

Managing Scripts

Create and Edit Scripts

Lua scripts are edited and created in a separate Script Editing Program. **Do not edit existing Lua scripts unless our technicians are walking you through the process.** To open an existing Lua script, double click on the script in the main list window. The script will open in the specified script editing program.

Most user scripts are created using the Script Editing **GUI**. To open an existing Gsf script, double click on the script in the main list window.

To create a new script, highlight a group or user name, right click, and select New. You will be asked to name the script and select a language. If you are not familiar with the Lua programming language, select Gsf and choose OK.



The new script will be opened in the appropriate Script Editing Module depending on the chosen format.

Import and Export Scripts

To import existing scripts, select the user/group that will own the script. Right click and choose import from the menu. A file browser will open. Find the script (either a Lua or Gsf script), select it, and choose Open. If the script is a Lua script, no further action is necessary. If it is a Gsf script, it will need to be opened, saved (compiled), and then closed.

Script Admin

To export an existing script, select the script from the main list window. Right click and choose export from the menu. A file browser will open. Find the export location, select it, and choose Open.

Route Calls to Scripts

Calls are routed to scripts via the telephony server routing tables found in the Telephony Services module (see the Telephony Services manual for more information). However, if you create a script and want to test it, you can invoke a script from the Text Call window (if you have administrative privileges). In the Call box, enter "**script:username/scriptname**" where *username* is the owner of the script and *scriptname* is the name of the script.

