e-speak

System Deployment Console User Manual

May 2000
Version 1.1

Table of Contents

Introduction

This is the user manual for the System Deployment Console. The System Deployment Console is a task-level configuration and management tool for e-speak. We will refer to it as “Console” throughout the rest of the manual.

The Console: Purpose

The purpose of the System Deployment Console is to:

- Help service operators to configure and deploy services remotely.
- Provide remote management at *task (process) level*.
- Help people to debug, test their services remotely.
- Allow easy and intuitive access to web-based management.

The Console: Basic features

1. Launching/ stopping e-speak cores, clients and native processes remotely.
2. Remote configuration tasks, command lines and options.
3. Remote monitoring of task execution states.
4. Easy creation of reusable scripts for creating system deployments.
5. Display remote task outputs. (You can use it to view any exceptions thrown by a remote core you just started, for example).
6. Configuration Management based on an object-relational model that describes not only individual settings but also relations between tasks and settings.
7. Persistent remote configuration. It automatically saves the configuration to the remote host and retrieves it upon connection and brings you the up-to-date status of the tasks **and** their relations.
8. Extensible CIM model based architecture. Service providers can implement custom management plug-ins for their own services and tasks and these become part of the Console Schema.
9. HTML based Data Pane that supports www-like navigation of management objects within the Console.
10. Integration with web-based management servlets and e-speak HTTP bridge.

The Console: Pre-requisites

Any remote machines that the Console is to manage must have a Daemon Core[1] resident. Connection to remote Daemon Cores requires either a local Daemon Core or a standard e-speak core. In order to view web-based management servlets you will need a standard web-browser on your machine.

Management Database Concepts

The Console is a remote management tool implemented in e-speak. It has been designed for the distributed management of both e-speak and native (non-e-speak) processes as well as services. The architecture of the Console is based on the Common Information Model (CIM) for system management and features many e-speak-specific extensions. The Console uses a locally stored database, essentially a table of *managed elements* representing, and interacting with real processes and services. Each element can have a number of *properties*, while each property has a series of *constraints* associated with them. For example, a "port number" property for e-speak core maybe have constraints that port number must be positive. The database also holds *associations*. Each type of association represents a particular kind of relationship between elements. Elements, associations, properties and constraints form our language for describing a managed system.

The Console has a built-in model for e-speak systems. It includes task-level management models for e-speak enabled hosts, cores, clients and services. Such a model forms the basis of the Console's inner-workings. This model can be extended and even replaced to allow custom system management. Normally, the users of the Console do not have to be aware of the existence of the model. Nevertheless, if you are interested in knowing the details, such as constraints applied to certain properties, the "Write Schema to File" feature of the Console allows you to save the content of the model into an XML file called the Management Schema File. The syntax and semantics of the Management Schema File is described in detail in "System Deployment Console: Developer Guide".

Each icon in the trees displayed in the left-hand side of the Console's interface represents a single instance of management element. The tree structure and hierarchy can be seen as a visual representation of the key associations in between the elements. By default, it shows the dependency hierarchy between elements.

[1] For more information on e-speak Daemon Core Service, see "e-speak Daemon Core Service User Manual".

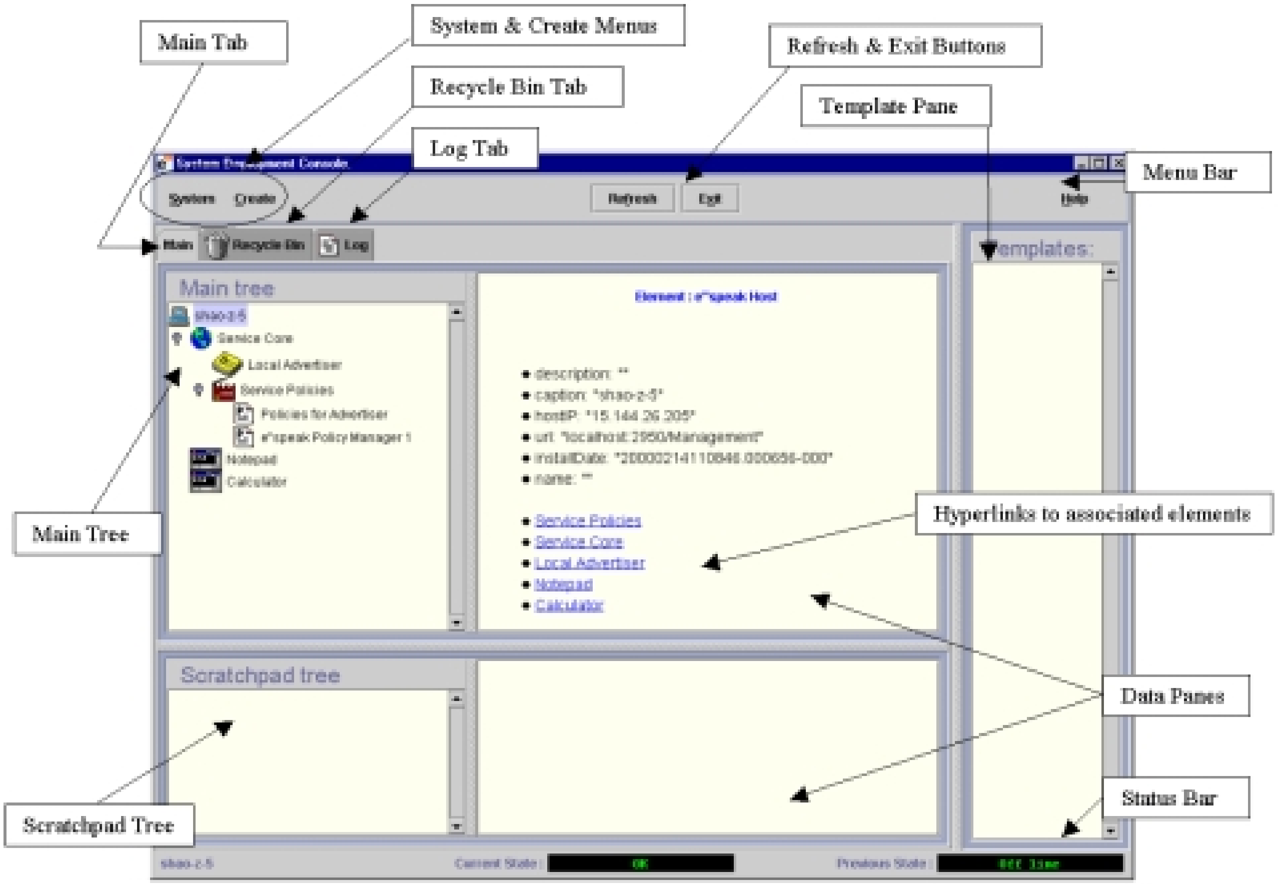
Layout

This section features labeled illustrations of the three different displays inside the main Console window. The three tabs labeled in Figure 1 (*Main Tab, Recycle Bin Tab, and Log Tab)*, enable the user to switch between these views. The items on the Menu Bar (also in Figure 1) are common to all three views. The Menu Bar contains buttons to refresh the screen (*Refresh Button*), and exit the Console (*Exit Button*). The main menus, the *Create Menu*, and the *System Menu* are all located here. The status bar is also common to all views though is only relevant to the Standard View.

Standard View

The standard view as shown below depicts the main managed deployment. It is the view you see when the Console gets started. When the 'Main Tab' is depressed, this view is enabled. The window contains the following:

- The Scratchpad Tree: where semi-configured items reside.
- The Main Tree: where fully configured components of the deployment reside.
- The Template Pane: where element templates (see: Templates) are contained.
- Two Data Panes (one for each tree window), containing HTML based information about the selected element in the corresponding tree.



Main tree

- shao-z-5
  - Service Core
    - Local Advertiser
    - Service Policies
      - Policies for Advertiser
      - e"speak Policy Manager 1
  - Notepad
  - Calculator

Element : e"speak Host

- description: ""
- caption: "shao-z-5"
- hostIP: "15.144.26.205"
- url: "localhost:2950/Management"
- installDate: "20000214110846.000656-000"
- name: ""

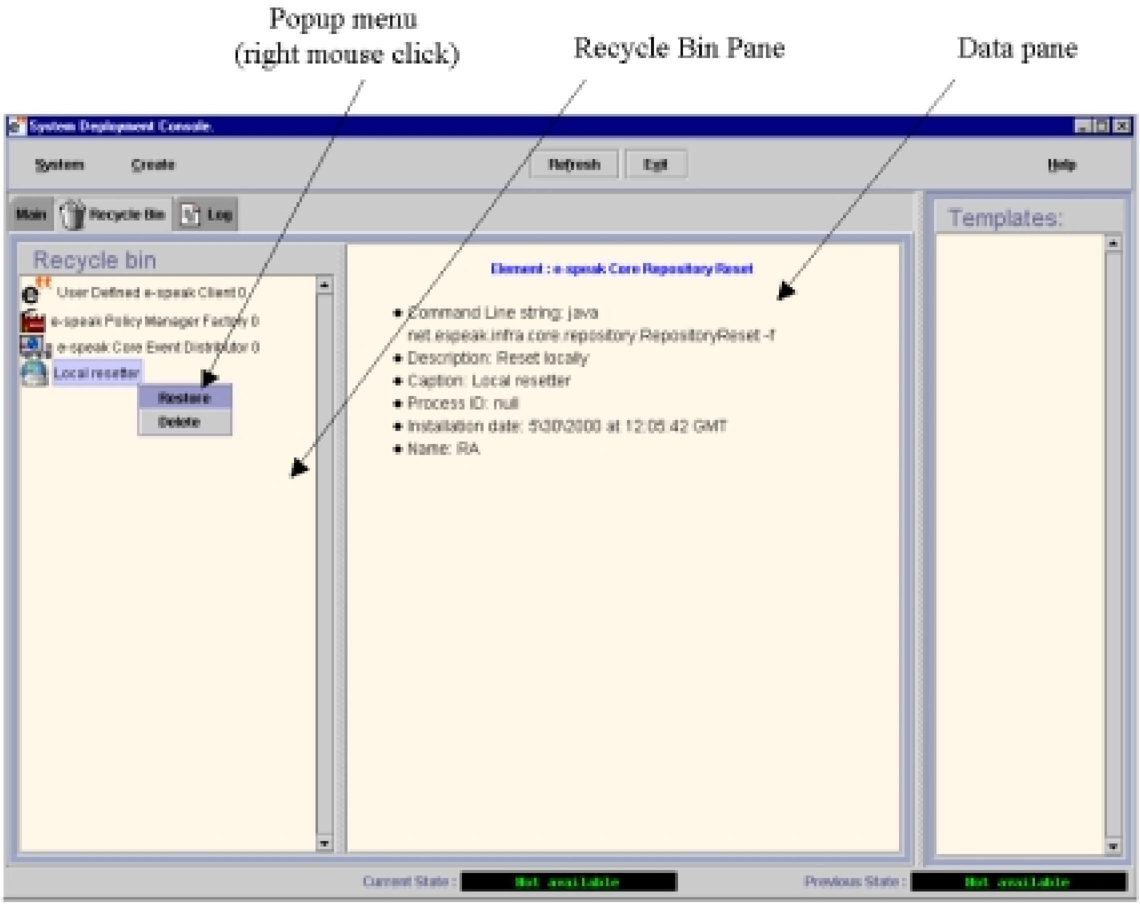
- Service Policies
- Service Core
- Local Advertiser
- Notepad
- Calculator

shao-z-5 Current State : OK Previous State : Off line

Figure 1 The standard view (Main Tab depressed)

Recycle Bin View

- The view when the Recycle Bin Tab is depressed. This view shows the processes in the *Recycle Bin*. See Recycle Bin for more details on the purpose and functionality of the Recycle Bin. The Data Pane displays the same HTML as with selected elements in the Standard View (above). The Status Bar is of no relevance to elements selected in this view.



Element : e-speak Core Repository Reset

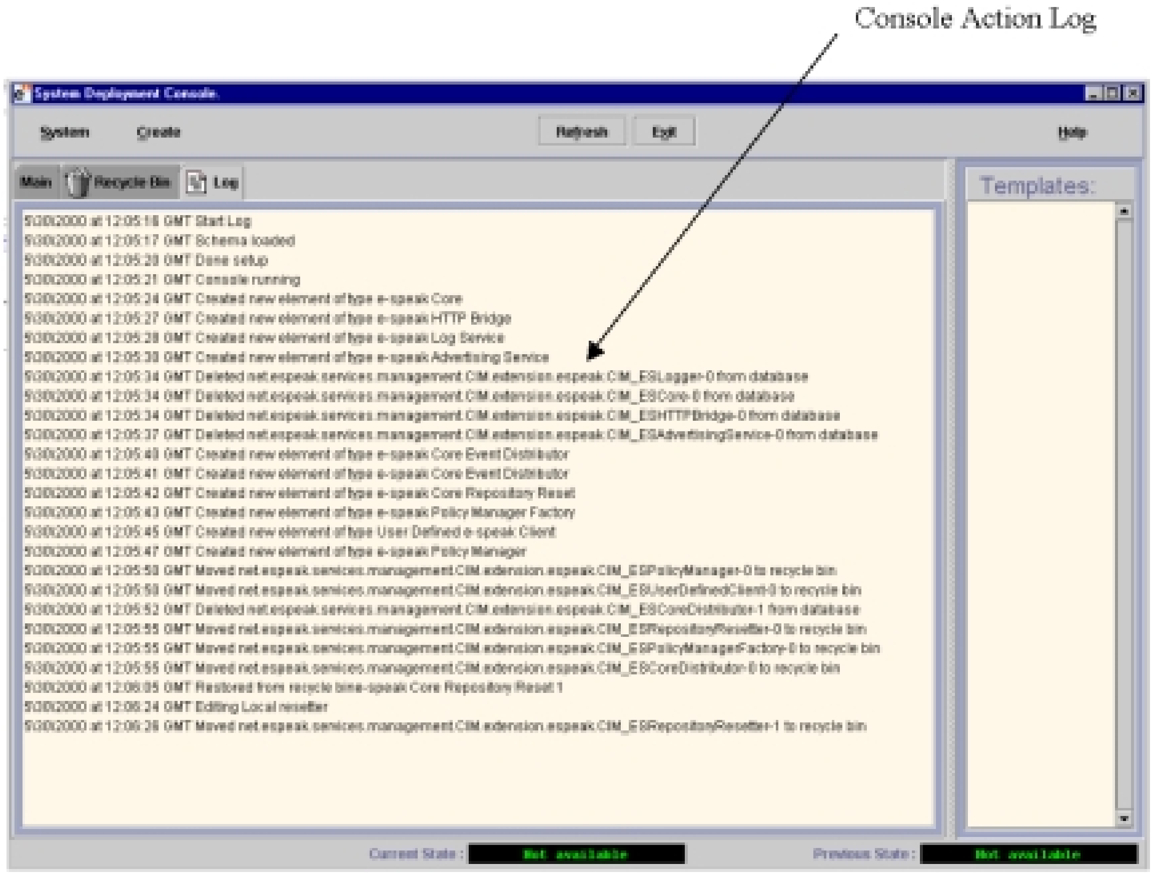
- Command Line string: java net.espeak.infra.core.repository.RepositoryReset -f
- Description: Reset locally
- Caption: Local resetter
- Process ID: null
- Installation date: 5\30\2000 at 12:05:42 GMT
- Name: RA

See Also:

Recycle Bin

Log view

- The view when the *Log* Tab is depressed. This view shows an active log of all user interactions with the Console. The Status Bar is of no relevance to this view.



```
5/30/2000 at 12:05:16 GMT Start Log
5/30/2000 at 12:05:17 GMT Schema loaded
5/30/2000 at 12:05:20 GMT Done setup
5/30/2000 at 12:05:21 GMT Console running
5/30/2000 at 12:05:24 GMT Created new element of type e-speak Core
5/30/2000 at 12:05:27 GMT Created new element of type e-speak HTTP Bridge
5/30/2000 at 12:05:28 GMT Created new element of type e-speak Log Service
5/30/2000 at 12:05:30 GMT Created new element of type e-speak Advertising Service
5/30/2000 at 12:05:34 GMT Deleted net.espeak.services.management.CIM.extension.espeak.CIM_ESLogger-0 from database
5/30/2000 at 12:05:34 GMT Deleted net.espeak.services.management.CIM.extension.espeak.CIM_ESCore-0 from database
5/30/2000 at 12:05:34 GMT Deleted net.espeak.services.management.CIM.extension.espeak.CIM_ESHTTPBridge-0 from database
5/30/2000 at 12:05:37 GMT Deleted net.espeak.services.management.CIM.extension.espeak.CIM_ESAdvertisingService-0 from database
5/30/2000 at 12:05:40 GMT Created new element of type e-speak Core Event Distributor
5/30/2000 at 12:05:41 GMT Created new element of type e-speak Core Event Distributor
5/30/2000 at 12:05:42 GMT Created new element of type e-speak Core Repository Reset
5/30/2000 at 12:05:43 GMT Created new element of type e-speak Policy Manager Factory
5/30/2000 at 12:05:45 GMT Created new element of type User Defined e-speak Client
5/30/2000 at 12:05:47 GMT Created new element of type e-speak Policy Manager
5/30/2000 at 12:05:50 GMT Moved net.espeak.services.management.CIM.extension.espeak.CIM_ESPolicyManager-0 to recycle bin
5/30/2000 at 12:05:50 GMT Moved net.espeak.services.management.CIM.extension.espeak.CIM_ESUserDefinedClient-0 to recycle bin
5/30/2000 at 12:05:52 GMT Deleted net.espeak.services.management.CIM.extension.espeak.CIM_ESCoreDistributor-1 from database
5/30/2000 at 12:05:55 GMT Moved net.espeak.services.management.CIM.extension.espeak.CIM_ESRepositoryResetter-0 to recycle bin
5/30/2000 at 12:05:55 GMT Moved net.espeak.services.management.CIM.extension.espeak.CIM_ESPolicyManagerFactory-0 to recycle bin
5/30/2000 at 12:05:55 GMT Moved net.espeak.services.management.CIM.extension.espeak.CIM_ESCoreDistributor-0 to recycle bin
5/30/2000 at 12:06:05 GMT Restored from recycle bine-speak Core Repository Reset 1
5/30/2000 at 12:06:24 GMT Editing Local resetter
5/30/2000 at 12:06:26 GMT Moved net.espeak.services.management.CIM.extension.espeak.CIM_ESRepositoryResetter-1 to recycle bin
```

Basic Operations

Launching the Console

The Console itself is an e-speak client. Like all e-speak clients, it must be connected to a local e-speak core to start.

The following command line starts the Console:

```
java net.espeak.services.management.configman.ui.Console <12345>
```
[2]

The Console will not be ready until the local core connection has been established or timed-out.

Connecting to E-speak Enabled Hosts

After the Console is started, the first thing one should do is to connect to the host that needs to be managed. Select the menu item *'System/Connect to Host'* as shown in Figure 2. The connect dialog box will appear (see Figure 3).

[2] 12345 is an example of the port number of your local daemon core (or a local standard e-speak core running a Connection Factory).

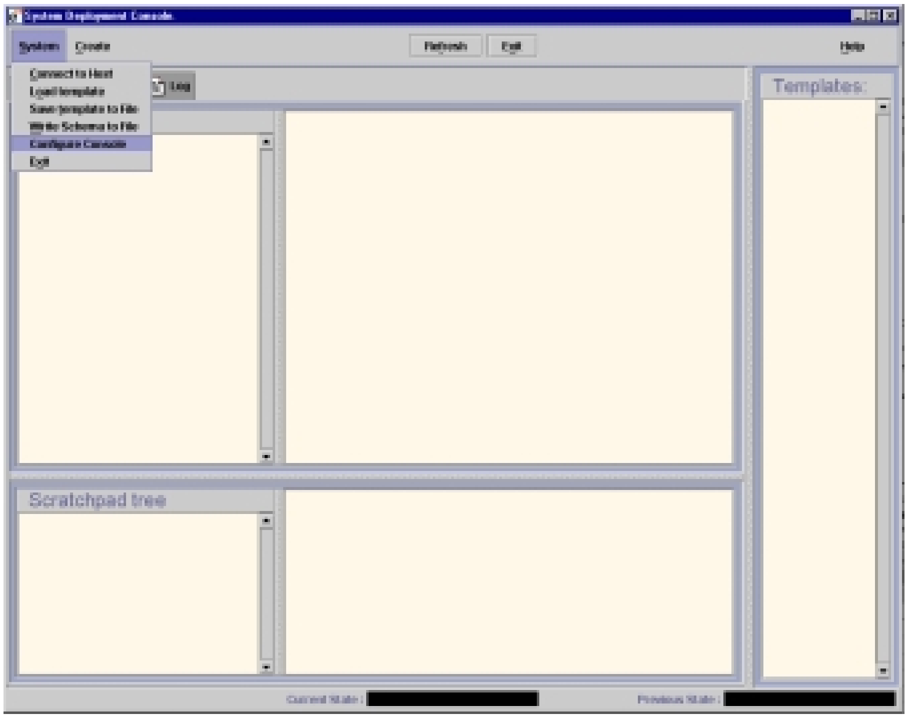
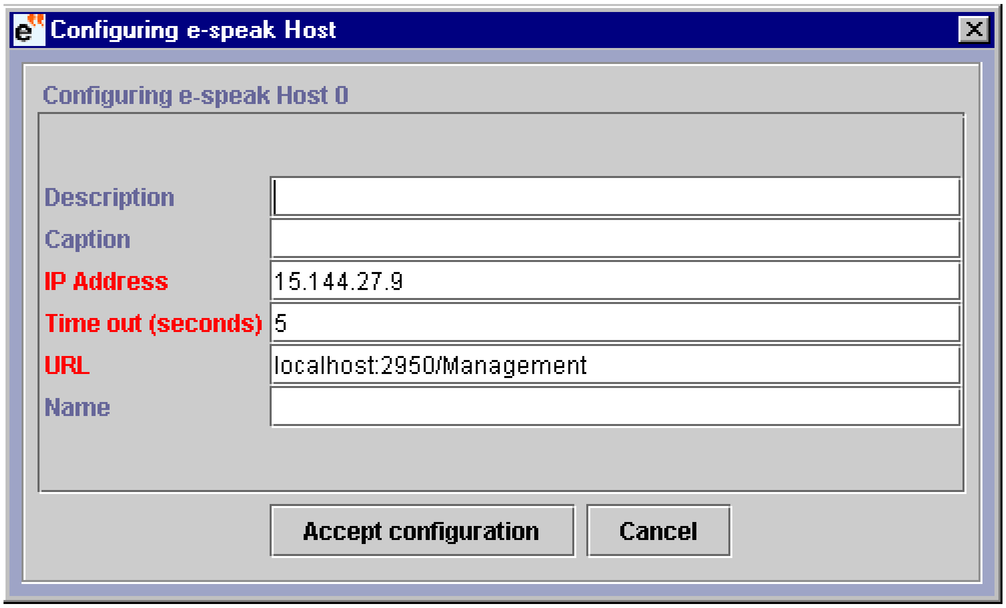


Figure 2 Click on "Connect to Host" on the "System" menu to connect to a remote host.

Figure 3 illustrates the “Configuring e-speak Host” dialog. Note: in all Console configuration dialog boxes, entries marked red are required. For the Host dialog, we need to enter values for the *hostIP* and *url* properties.

You should fill in the IP address of the host that you want to manage in the hostIP property.

The “url” field shall contain the URL of the Advertising Group at which the host management service can be found. For example, in Figure 3, the *url* field contains the value “localhost:2950/Management”. This value points to an advertising service at port 2950, host “localhost”, and with advertising group name “Management”. The Console uses the *url* field to set the context of lookup for the remote host management service. For more information about the Advertising Service and Advertising Groups, see the “e-speak Advertising Service usage document”. Note that if the remote host uses the Daemon Core program to provide the host management service (as by default), the port number is fixed to 2950.



Configuring e-speak Host

Configuring e-speak Host 0

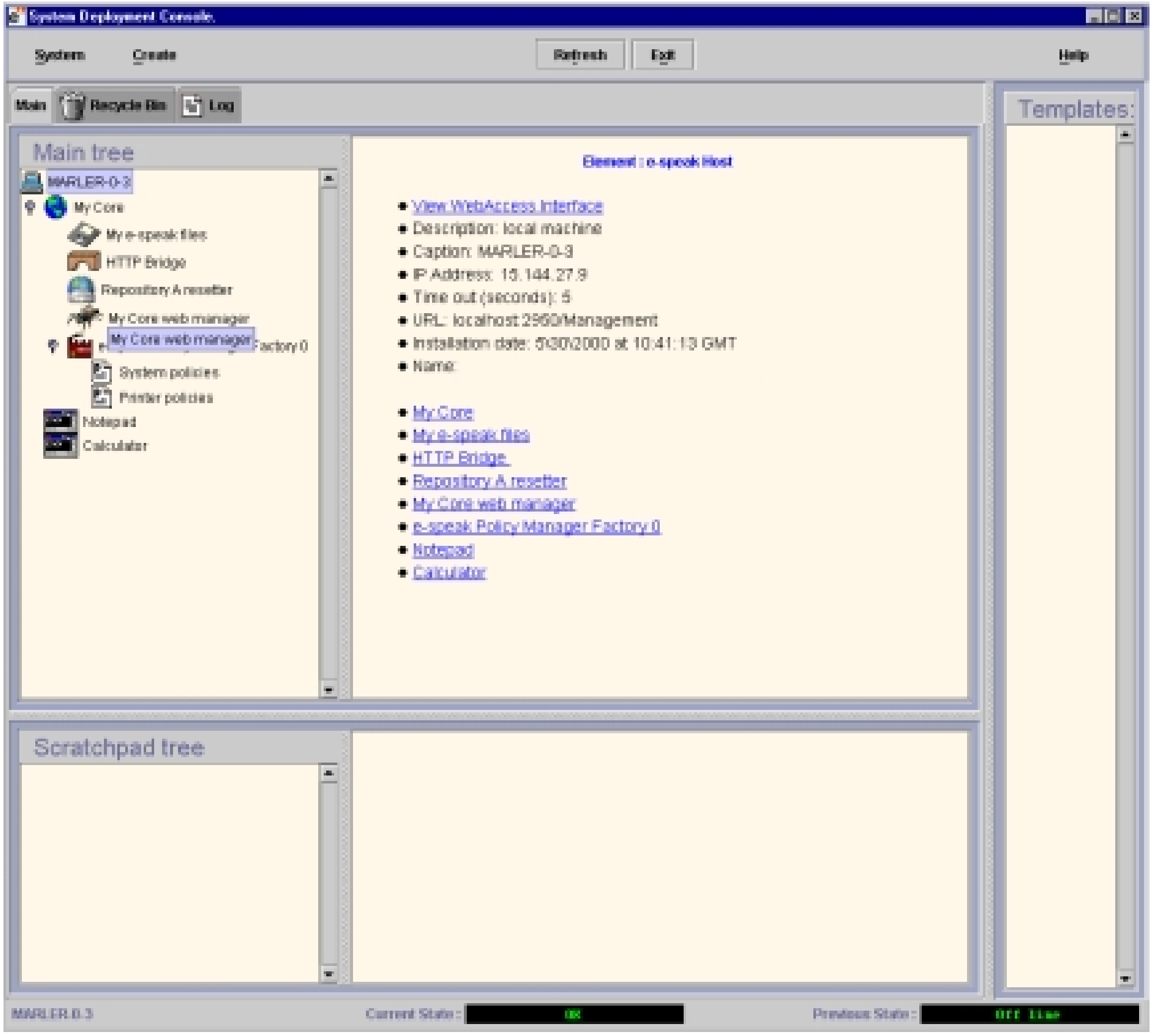
Description	
Caption	
IP Address	15.144.27.9
Time out (seconds)	5
URL	localhost:2950/Management
Name	

Figure 3 The "Connect" Dialog Box

Once the form has been completed, click '*Accept configuration*'. If the connection is successful, a host object and all its hostees will be displayed in the Tree Pane of the Main Panel as in Figure 4.

By left clicking on that host object, you will see a list of properties about that host appear in the Data Pane. At the bottom of the Console, on the status bar, the current state of the host is shown to be OK, so it is ready to be managed. You may now create and configure tasks on that host. Right click on the host object in the Tree Pane to bring up a popup menu showing what you can do with that object. It usually includes operations for creating, starting, and stopping tasks, editing properties, etc. Selecting "Properties" will bring up the "Configuring e-speak Host" dialog again. Here you may wish to assign a 'description' and 'caption' to that host, for example. The required fields are marked in red.

If there are already managed objects (such as tasks, cores, and clients) created on a host, connecting to that host will automatically retrieve the management information about those elements and they will be displayed on the Tree Pane. This is illustrated in Figure 4.



Element : e-speak Host

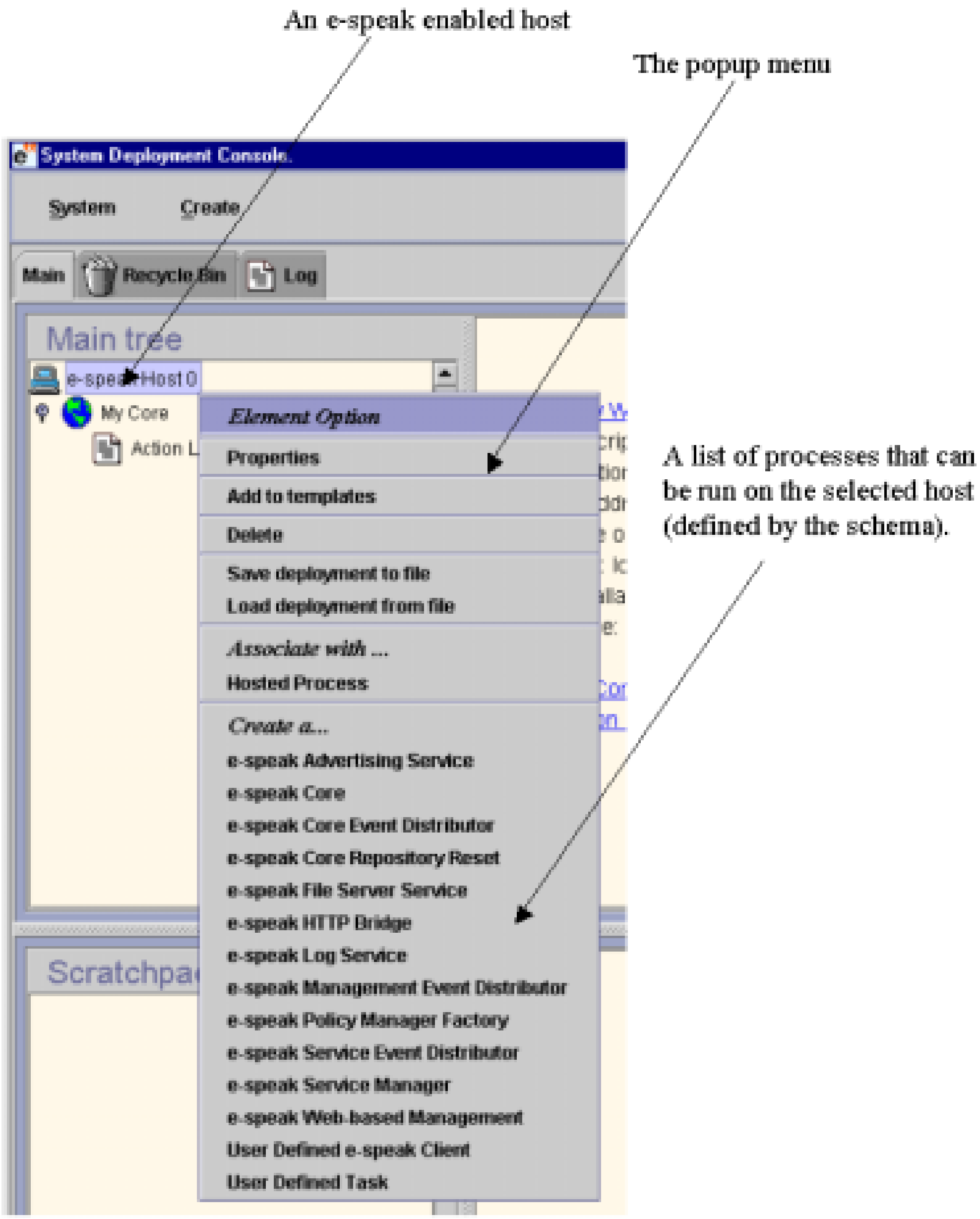
- View WebAccess Interface
- Description: local machine
- Caption: MARLER-0-3
- IP Address: 15.144.27.9
- Time out (seconds): 5
- URL: localhost:2950/Management
- Installation date: 5\30\2000 at 10:41:13 GMT
- Name:

- My Core
- My e-speak files
- HTTP Bridge
- Repository A resetter
- My Core web manager
- e-speak Policy Manager Factory 0
- Notepad
- Calculator

Figure 4 Connecting to a host with processes on it will automatically display them in the tree.

Creating Managed Elements

Once connected to an e-speak enabled host a computer icon will appear in the top left-hand section of the console. By right-clicking on this icon a popup menu appears (see Figure 5 The popup menu). The items on this menu represent operations that can be done to, or done with, this element. At the bottom of this menu, under the heading '*Create a…*' a list of elements that can be started on this host is displayed. By clicking on one of these process types a Configuration Form for the new process will be displayed (for more information on the Configuration Form see *Configuring Element Properties*). Once all required properties are set for the new element, it will appear in the same window as the host. This element is now in a manageable state.



System Deployment Console
System
Create
Main
Recycle Bin
Log
Main tree
e-speak Host 0
My Core
Action L
Scratchpa
Element Option
Properties
Add to templates
Delete
Save deployment to file
Load deployment from file
Associate with ...
Hosted Process
Create a...
e-speak Advertising Service
e-speak Core
e-speak Core Event Distributor
e-speak Core Repository Reset
e-speak File Server Service
e-speak HTTP Bridge
e-speak Log Service
e-speak Management Event Distributor
e-speak Policy Manager Factory
e-speak Service Event Distributor
e-speak Service Manager
e-speak Web-based Management
User Defined e-speak Client
User Defined Task

An e-speak enabled host

The popup menu

A list of processes that can be run on the selected host (defined by the schema).

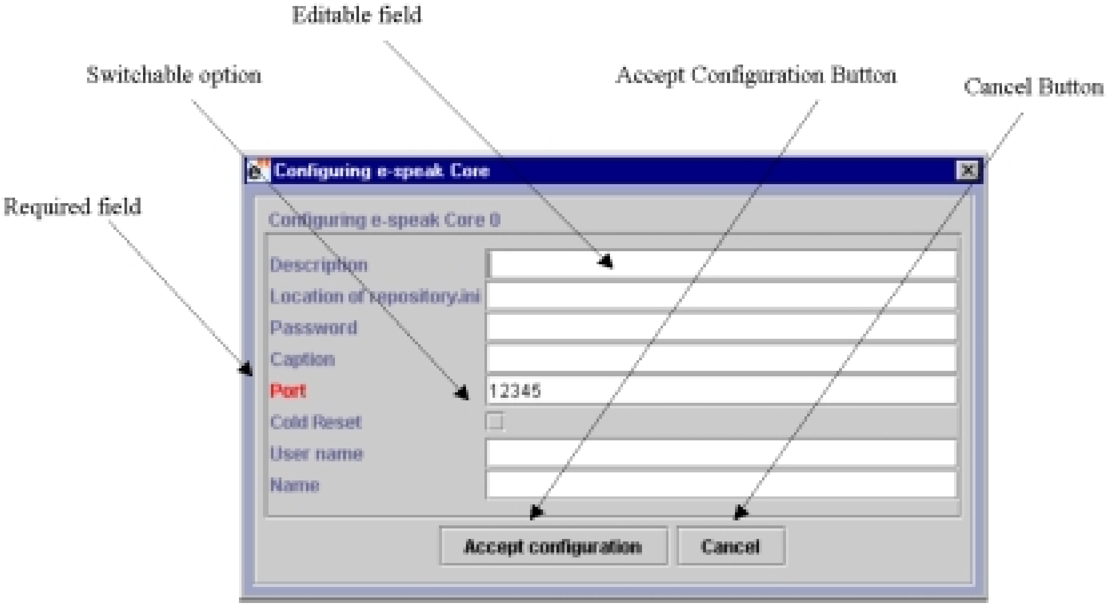
Figure 5 The popup menu.

Configuring Element Properties

By right-clicking on the element and selecting '*properties*', a Configuration form, illustrated in Figure 6, will appear. It allows you to change editable properties of the element.

Most common you will see three properties "description", "name", "caption". These properties are used by the user customize labeling and grouping of elements. None of these properties are mandatory. By default, the element will be displayed with its caption. If the caption field is blank, a default label will be given, which consists of object's type plus a unique integer ID.

There are other properties that allow you to configure elements. The e-speak core, for example, requires a *port* property (it is mandatory as it is marked in red). The value in this field must be a valid e-speak port number; the Console will check the validity of the properties using pre-defined constraints. If an invalid property value is entered, a message will be displayed to prompt you for re-entry. After clicking the *'Accept configuration'* (see Figure 6) button at the bottom of the form the configuration will be set for the element if there are no errors.



Configuring e-speak Core

Configuring e-speak Core 0

Description
Location of repository.ini
Password
Caption
Port 12345
Cold Reset □
User name
Name

Figure 6 Configuring task properties.

If some mandatory fields are left empty, the element will be displayed in the 'Scratchpad Tree' (as the Element is ill-configured). Once all mandatory fields have been correctly entered into the configuration form, the element will be displayed in the 'main tree' (see *Layout definitions*).

Starting Elements

Certain elements such as cores, clients, processes, and services, once correctly configured, can be started and stopped. To start an element, bring up its popup menu (right-clicking on the selected process) and select the 'Start' option (see Figure 5). If the element was not correctly configured and the start command could not be carried out, a message dialog will appear. The 'Previous State' and 'Current State' information on the status bar (see *Monitoring Element Status*)allows you to monitor the state of started elements.

Stopping Elements

A running element can be stopped using the 'Stop' option from popup menu (see Figure 5).

Removing an Element from the Configuration

The popup menu for every element has a "Delete" option[3](see Figure 5). If this is selected the element will be moved to the Recycle Bin. Note that delete doesn't attempt to stop an active element, so an element in the Recycle Bin can still be running until you stop it. For more information see *Recycle Bin Chapter*.

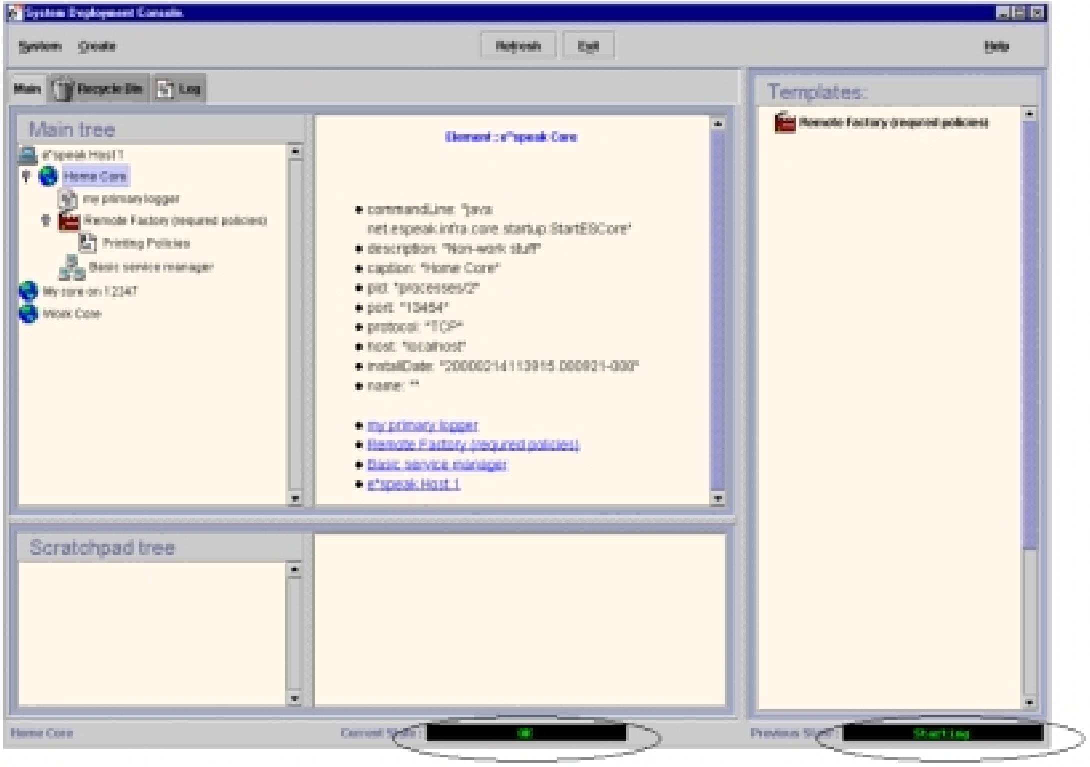
[3] This operation is also accessible by pressing <Delete> when a process in either of the left-hand windows is selected.

Monitoring the Status of an Element

There are two ways for monitoring the running status of elements within the System Deployment Console, the status bar at the bottom of the Console and the standard output dialog box.

Status Bar

The status bar at the bottom of the Console shows the current and previous state of the element last selected in the user interface. In the example below, Figure 7, a user-defined task is highlighted. The status bar indicates that the current state of the task is ‘OK’ and the previous state of the task is ‘Starting’.



Element : e"speak Core

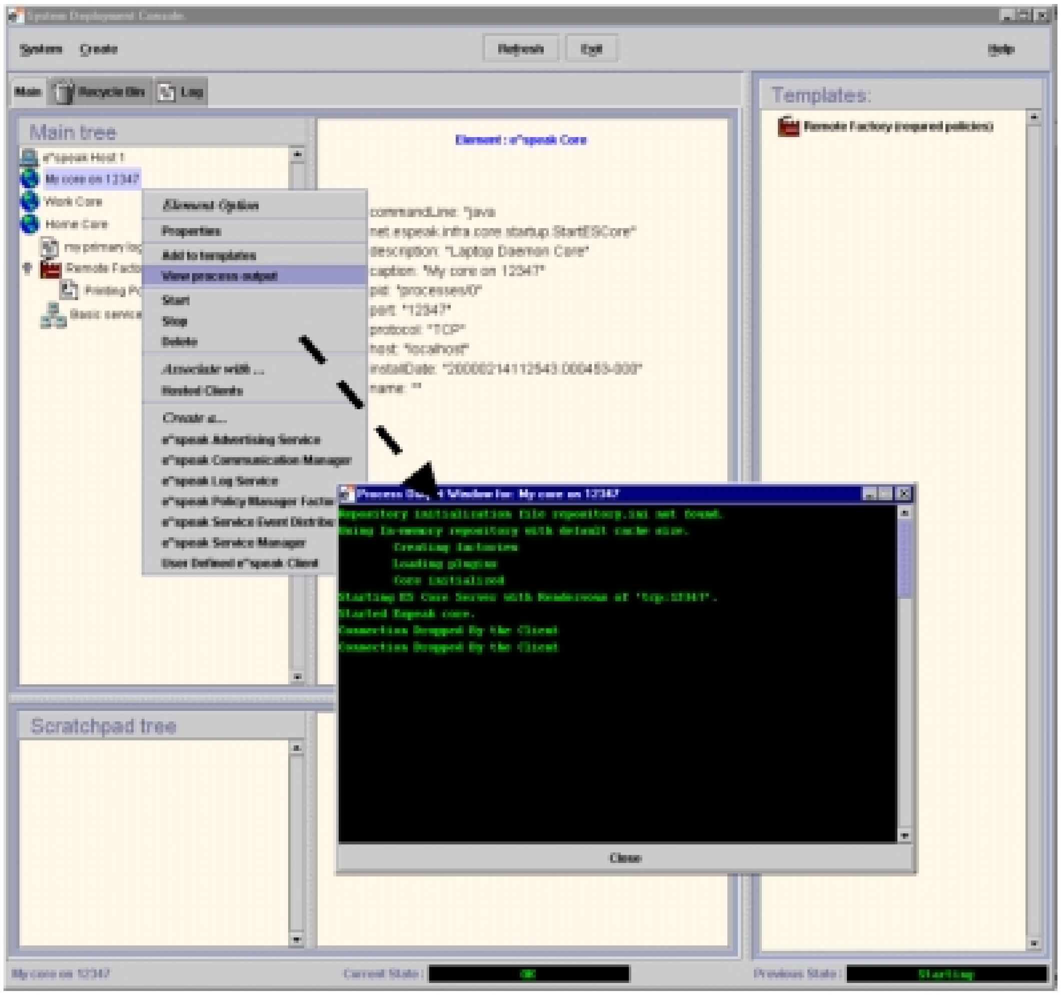
- commandLine: "java net.espeak.infra.core.startup.StartESCore"
- description: "Non-work stuff"
- caption: "Home Core"
- pid: "processes/2"
- port: "13454"
- protocol: "TCP"
- host: "localhost"
- installDate: "20000214113915.000921-000"
- name: ""

- my primary logger
- Remote Factory (required policies)
- Basic service manager
- e"speak Host 1

Figure 7 Checking the running state of an object.

Standard Output

For more information about the element’s execution state, it is possible to view the standard output for a process element. To view the standard output select an object from the main panel of the console, click the right mouse button and from the popup menu select ‘View process output’.



commandLine: "java
net.espeak.infra.core.startup.StartESCore"
description: "Laptop Daemon Core"
caption: "My core on 12347"
pid: "processes/0"
port: "12347"
protocol: "TCP"
host: "localhost"
installDate: "20000214112543.000453-000"
name: ""

```
Repository initialization file repository.ini not found.
Using In-memory repository with default cache size.
        Creating factories
        Loading plugins
        Core initialized
Starting ES Core Server with Rendezvous at 'tcp:12347'.
Started Espeak core.
Connection Dropped By the Client
Connection Dropped By the Client
```

Figure 8 Viewing the standard output of a process.

Creating and Using Templates

In the Console it is possible to create element templates. Templates are partially configured elements that can be saved, loaded, reconfigured, and added into the current deployment. Templates are designed to be an easy way of storing and creating multiple instances of frequently used element configurations.

Creating a Template

To create a template from an object, select it from the main panel of the console, click the right mouse button and from the popup (see *The Popup Menu*) select *'Add to templates'*. The object is added to the template section on the right hand side of the Console (see Figure 9).

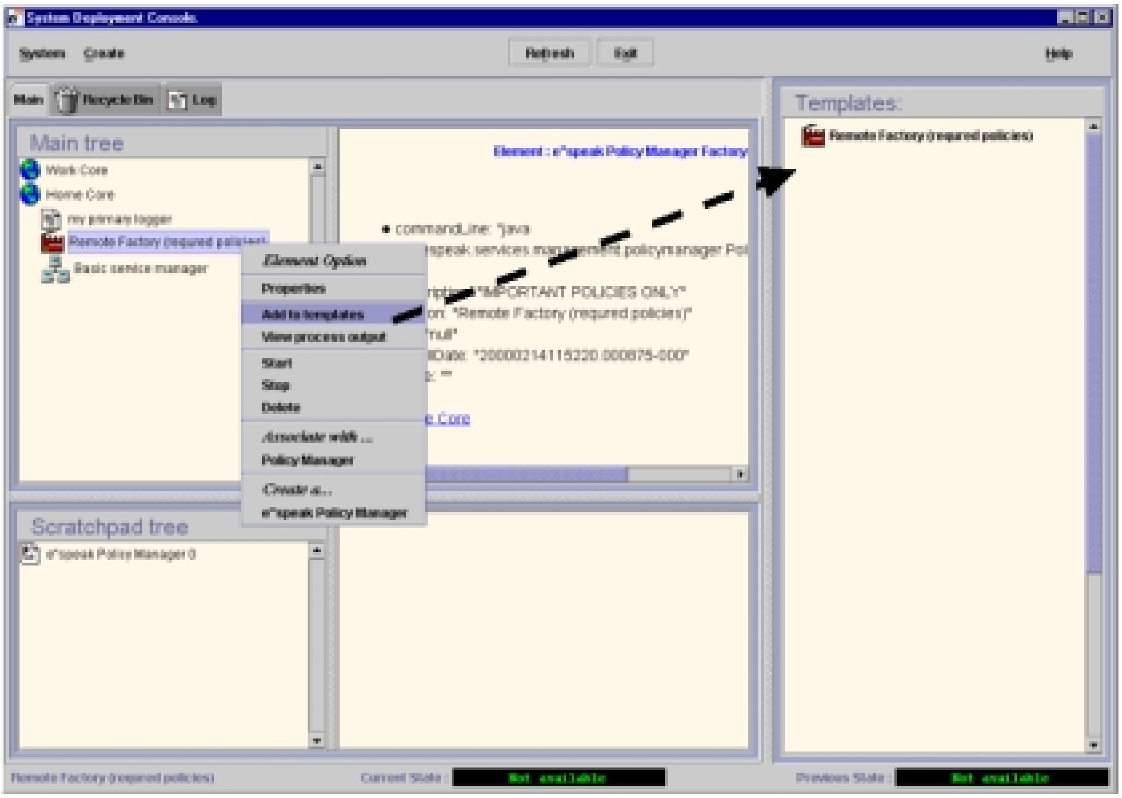


Figure 9 Adding An Object To The Templates.

Using a Template to Create an Element

Click the right mouse button on an object in the template pane and select 'Create' from the popup menu. An instance of the template object will be created in the main panel of console. Also, use the popup menu to delete the template object or to change its properties using the *Property Form*.

Saving the Current Set of Template Elements

From the menu bar at the top of the Console select *'System / Save Templates To File'*. Select a location and a file name, it is suggested that an '.esc' extension be used to distinguish it as an e-speak system configuration file.

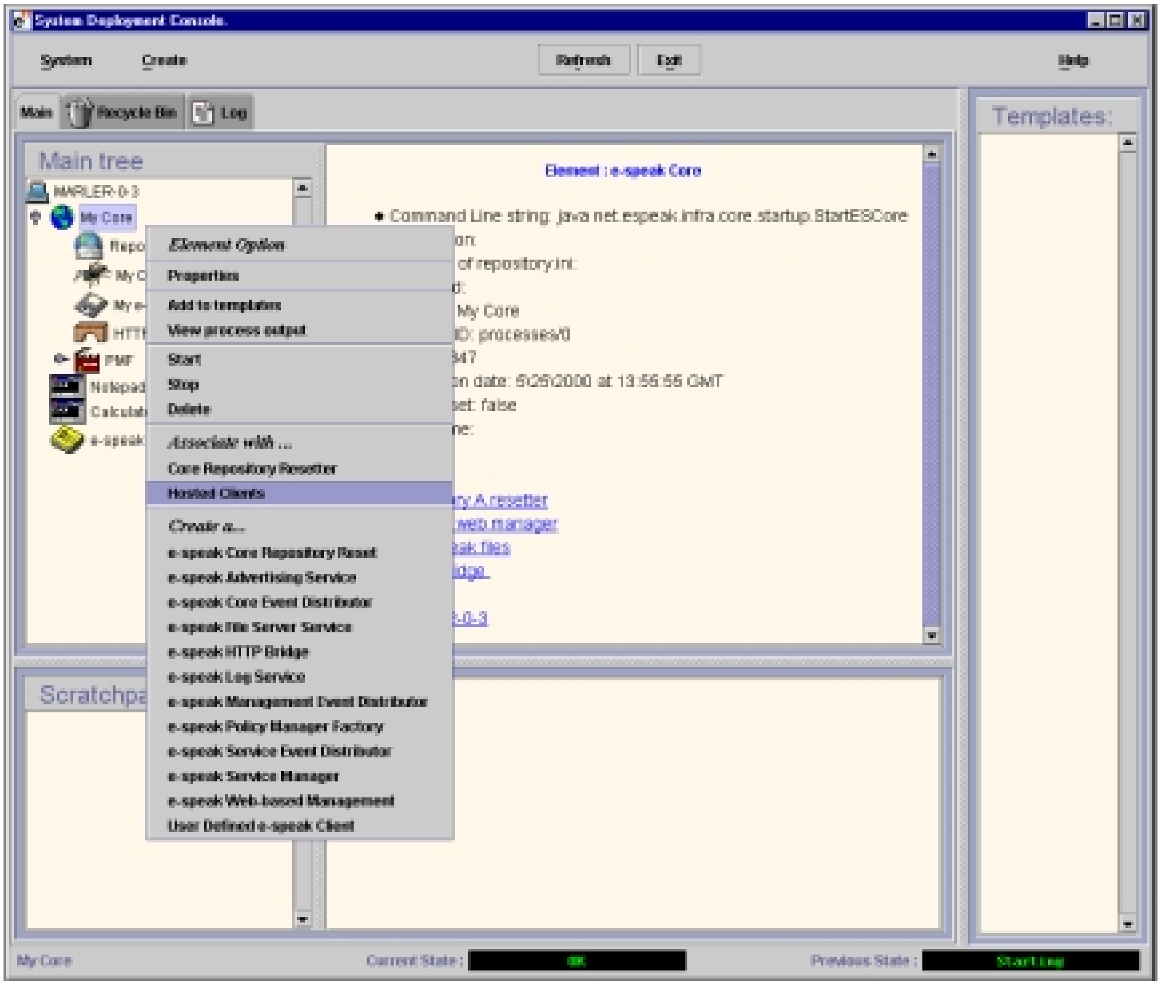
Loading a Template Set

From the menu bar at the top of the Console select *'System/ Load Template from File'*. On selecting a file and confirming, the elements in the template file will be added to the template pane.

Associating Elements with each other

Associations between elements are vital in system management (see Database Concepts). Some operations in the Console automatically generate these associations; others require the user to explicitly specify them. This section introduces the tool for creating associations in the Console.

Selecting an object and choosing to *'Create a…'* (task name) from its popup menu automatically creates an association between the existing and the new object about to be created. However, it is sometimes necessary to create associations between two existing objects explicitly. Such an association is created using the popup menu for the object you wish to associate. Select an option from the *'Associate With…'* section of the popup menu (Figure 10) and an Association Creator Dialog Box will appear.



Element : e-speak Core

- Command Line string: java net espeak infra core startup StartESCore

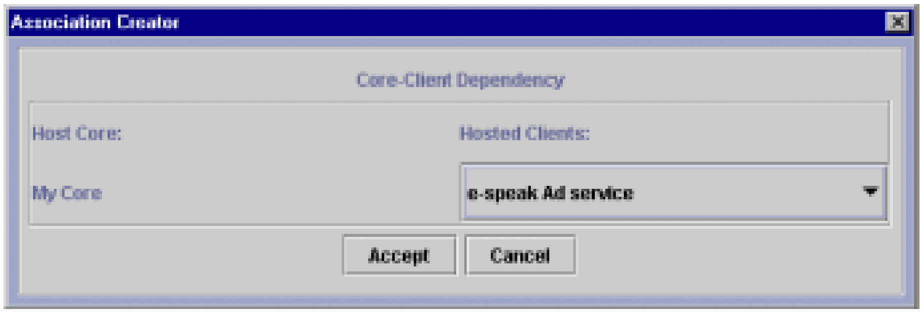
on:
of repository.ini:
d:
My Core
D: processes/0
347
on date: 5/25/2000 at 13:56:55 GMT
et: false
ne:

ry A resetter
web manager
ak files
dge
-0-3

Figure 10 Associating an e-speak core with an e-speak advertising service.

Creating Associations explicitly

After selecting “Associate with…“ on the core’s popup menu a dialog (Figure 11) pops up:



Host Core: My Core

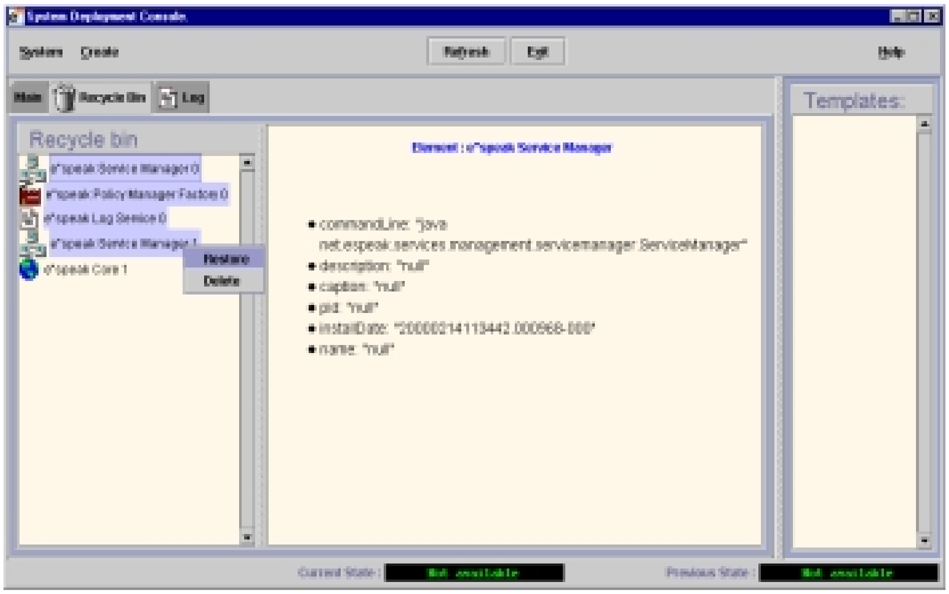
Hosted Clients: e-speak Ad service

Figure 11 Association Creator Dialog Box.

The drop-down box contains a list of all Elements in the database that would make appropriate associates. Select one and click accept to create the association.

Recycle Bin

When an object is deleted, it is moved to the Recycle Bin Pane. The Recycle Bin is a temporary, non-persistent store for elements that have been removed from the main configuration. No operations can be performed on these elements in the bin except restoration (re-attachment to the main configuration), or deletion (stopping and permanently removing the element). Items placed in the Recycle Bin lose any relations to other elements that they had whilst they were being managed. If an element is restored from the Recycle Bin, associations to other managed processes may need to be recreated in order to perform certain operations on it. An object can be deleted to the recycle bin and restored without affecting the elements to which it refers.



Element : e"speak Service Manager

- commandLine: "java net.espeak.services.management.servicemanager.ServiceManager"
- description: "null"
- caption: "null"
- pid: "null"
- installDate: "20000214113442.000968-000"
- name: "null"

Figure 12 Restoring Objects From The Recycle Bin.

Restoring Elements

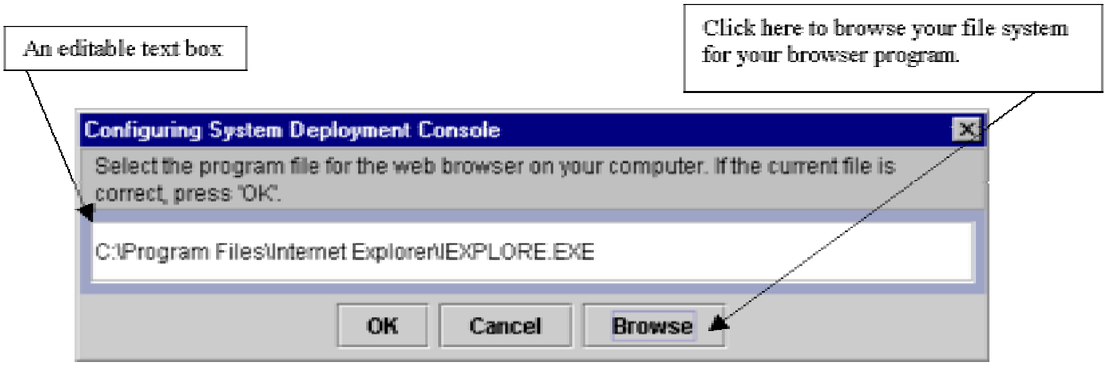
To restore a deleted object click on the Recycle Bin Tab, click on the appropriate object and click the right mouse button to bring up a pop-up menu. This menu has the 'Restore' and 'Delete' options (see Permanently Deleting Elements for more details). These operations are also available through shortcut keys ('Delete' performs deletion, 'CTRL-Enter' performs restoration). Several objects can be deleted or restored simultaneously by selecting multiple objects in the Recycle Bin. Pressing the <CTRL> key and selecting the required objects with the mouse can achieve multiple selections. To select all the objects in the Recycle Bin press ‘CTRL-A’.

Permanently Deleting Elements

Permanent deletion of elements bypasses the Recycle Bin and removes any reference to the task stored in the Console. Permanent deletion also forces the running tasks to be stopped. If you wish to discontinue the management of a task but do not wish to stop it yet, move the item to the recycle bin.

Configuring the Console with your web-browser

Some services that can be started with the Console also create servlets that can be used to manage the service remotely. The Console has been designed with the need to access these servlets in mind. Servlets are not viewable within the Console window as they may contain any web page content supported by new browser technologies. The Console instead can be configured to load the servlet web page into a browser of your choice. To do this configuration, click on the *‘System/ Configure Console’* menu option (or press <ALT>+<S> then <C>). This action will bring up the configuration window for the Console (see Figure 13).



An editable text box

Click here to browse your file system for your browser program.

Configuring System Deployment Console

Select the program file for the web browser on your computer. If the current file is correct, press "OK".

C:\Program Files\Internet Explorer\IEXPLORE.EXE

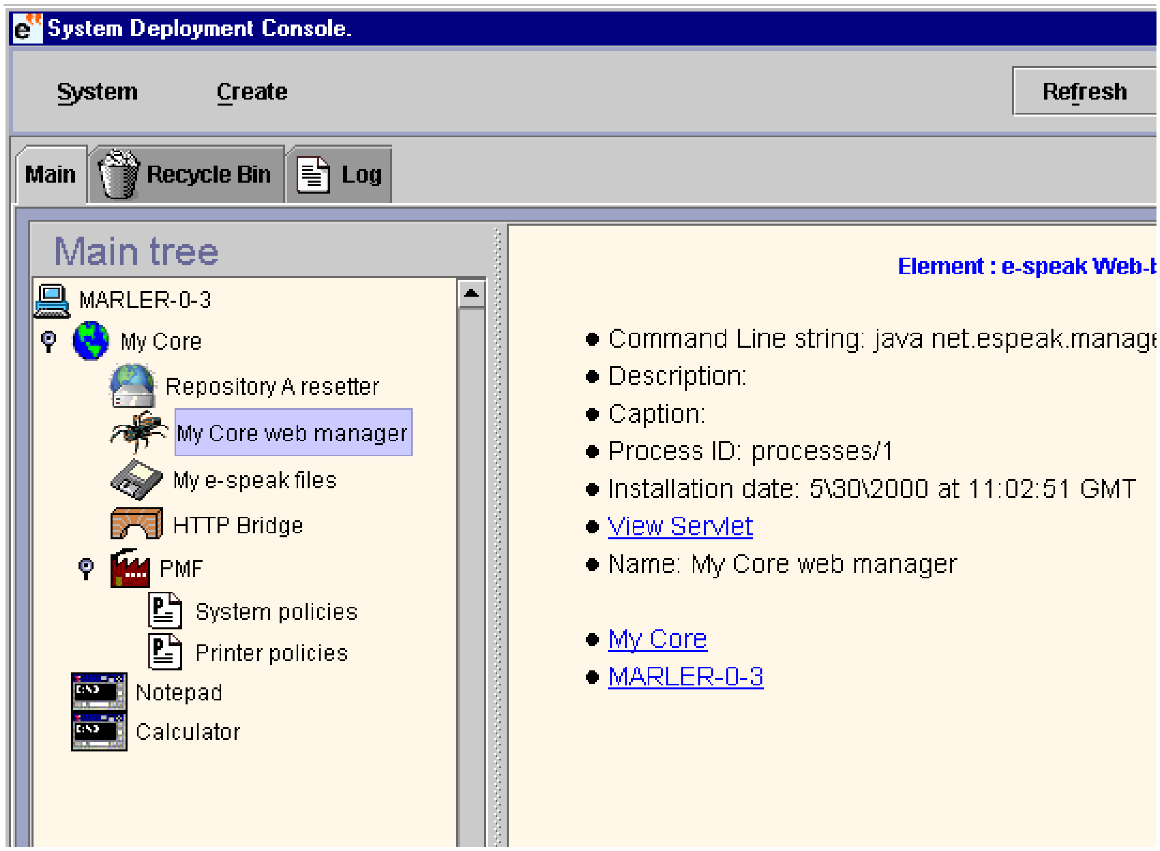
Figure 13 The Console Configuration Window

Simply type in the path to your browser program into the text box or browse you file system by clicking the *‘browse’* button. If you are unsure where the program resides on your computer it may be useful to look at the properties of the desktop icon you usually start your browser with. Once you are happy with your selection press *'OK'.*

The information entered into this window is stored persistently. This means that unless the Console's configuration file is deleted or corrupted this information will be remembered indefinitely (though the settings are still editable).

Viewing Servlets

Those e-speak services that are manageable via servlets have an extra hyperlink in their data pane. When clicked, the link (which reads View Servlet) will automatically load the servlet into the browser selected when configuring your Console (see Configuring the Console). The following picture shows the servlet's hyperlink in the data pane of a "Web-based management service":



- Command Line string: java net.espeak.manage
- Description:
- Caption:
- Process ID: processes/1
- Installation date: 5\30\2000 at 11:02:51 GMT
- View Servlet
- Name: My Core web manager

- My Core
- MARLER-0-3

Figure 14 An example of a service running a servlet.

Some of the servlets generated by e-speak services are “served” by an e-speak HTTP Bridge Service. This service can be started in the Console. Once started the servlets for these services will become active. If you find that View Servlet is a dead-link the most likely cause is that, though the service is generating the HTML for the servlet, without the HTTP Bridge running this information is not being served. Starting the e-speak HTTP Bridge will enable the View Servlet link. Also, the e-speak HTTP bridge does not serve files. To do this you will need to run the e-speak File Server Service. Running this service will also serve any images in the servlet web-pages. The e-speak Web-based Management Service is particularly useful as it auto-detects other services being served by the HTTP Bridge. Viewing the servlet for this service will provide you with an index of the servlets for all web-manageable services served by the bridge.

See Also:

Configuring the Console

Saving System Deployments as Reusable Scripts

The Console enables you to save a deployment of Elements under a host as a reusable script. These scripts can then be started on any hosts, on top of their existing deployments. Any Elements, Associations, and Properties that exist on a "live" deployment being saved will be reconstructed upon the re-deployment of the script. To save a deployment right-click on the e-speak enabled host. From the popup menu select *'Save deployment to file'*[4]. You will be presented with a file chooser window allowing you to select an existing file (or if you type in a file name that doesn't exist the file will be created. All deployment files have the .dply extension.

Once saved the deployment can immediately be loaded onto any host in the Main tree window. To load a configuration onto a host, right-click onto that host and choose *'Load configuration from file'*. After requesting confirmation for this action the Console will the load the deployment onto the selected host. The new deployment will not overwrite any existing hostees. Currently the Console does not start any of the new Elements. This behavior may be supported in later releases.

[4] This option is only available from the popup menus belonging to hosts.

The System Deployment Console Tutorial

Introduction

This chapter contains a step-by-step demonstration of the System Deployment Console illustrating many of its key features. It will be useful to read the user guide before attempting to follow the examples outlined here.

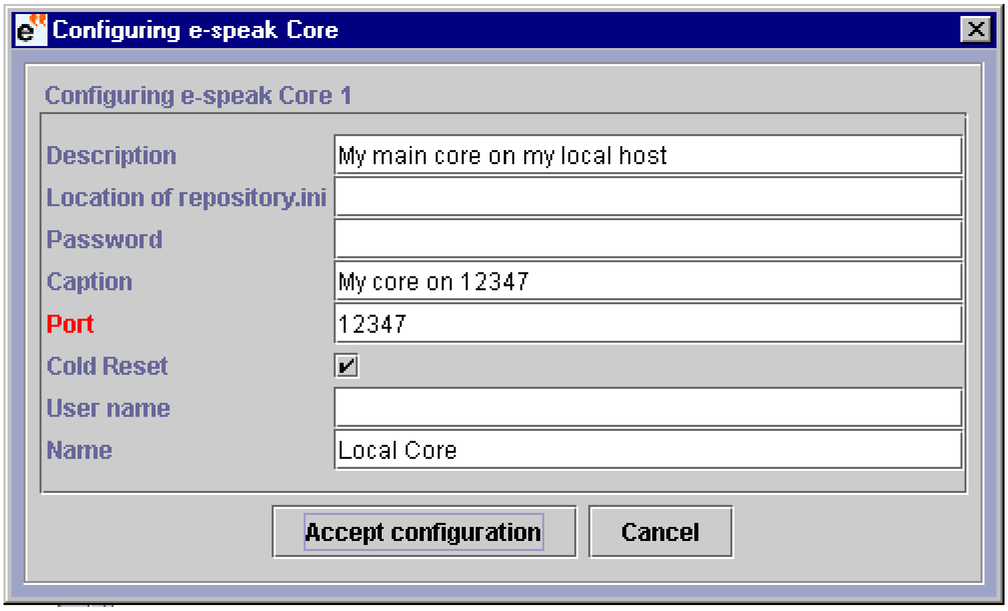
Step one – Starting the Console.

- Ensure that a Daemon Core is running on the local machine.
- Start a Console: "java net.espeak.services.management.configman.Console <*port_no*>"[5]
- Once the Console starts up open the *'System'* menu on the main menu bar. Choose the *'Connect to host'* option.
- Configure the host in the form provided.

Step two – Testing the connection

- Right-click on the host icon. From the *'Create a...'* menu select the *'e-speak core'* option. Configure the core the run on port number 12347 (the host field can stay as the default – localhost). Set the caption to "My core on 12347". Set the description to "My main core on my local host". Figure 15 illustrates how the configuration form should be filled in for this core.

[5] *port_no* the port number for the Daemon Core.



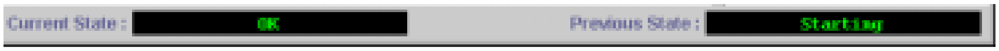
Configuring e-speak Core 1

Description	My main core on my local host
Location of repository.ini	
Password	
Caption	My core on 12347
Port	12347
Cold Reset	☑
User name	
Name	Local Core

Figure 15 The configuration form for the new core.

- Once configured, right-click on the core's icon in the main tree. Select *'Start*' from the popup menu.

- After a short time, the *'Current State'* field on the status bar should change from *Not Available* to '*OK*' (see Figure 16). If this occurs, the connection has been successful. To check the core started correctly you can select *'Process Output'* from its popup menu. If the message in the core's output window does not report an error then this test is complete.



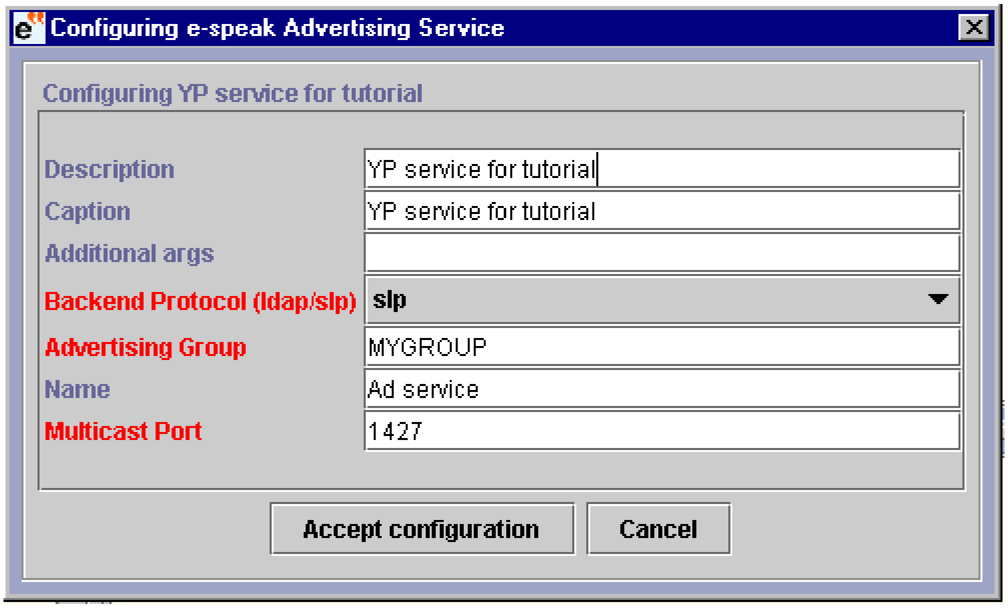
Current State : OK

Previous State : Starting

Figure 16 The status bar indicating running process.

Step three – Creating a e-speak service deployment

- In this step, we are going to run an '*e-speak Advertising Service'* on the new core (created in step two).
- Using the *'Create'* menu on the top menu bar select '*e-speak Advertising Service'*. As the service requires configuration it will appear in the Scratchpad tree.
- Configure the service in the Property Form (see Figure 17). Label it and enter a group name (e.g. "MYGROUP"). Select the 'slp' option for the backend protocol.

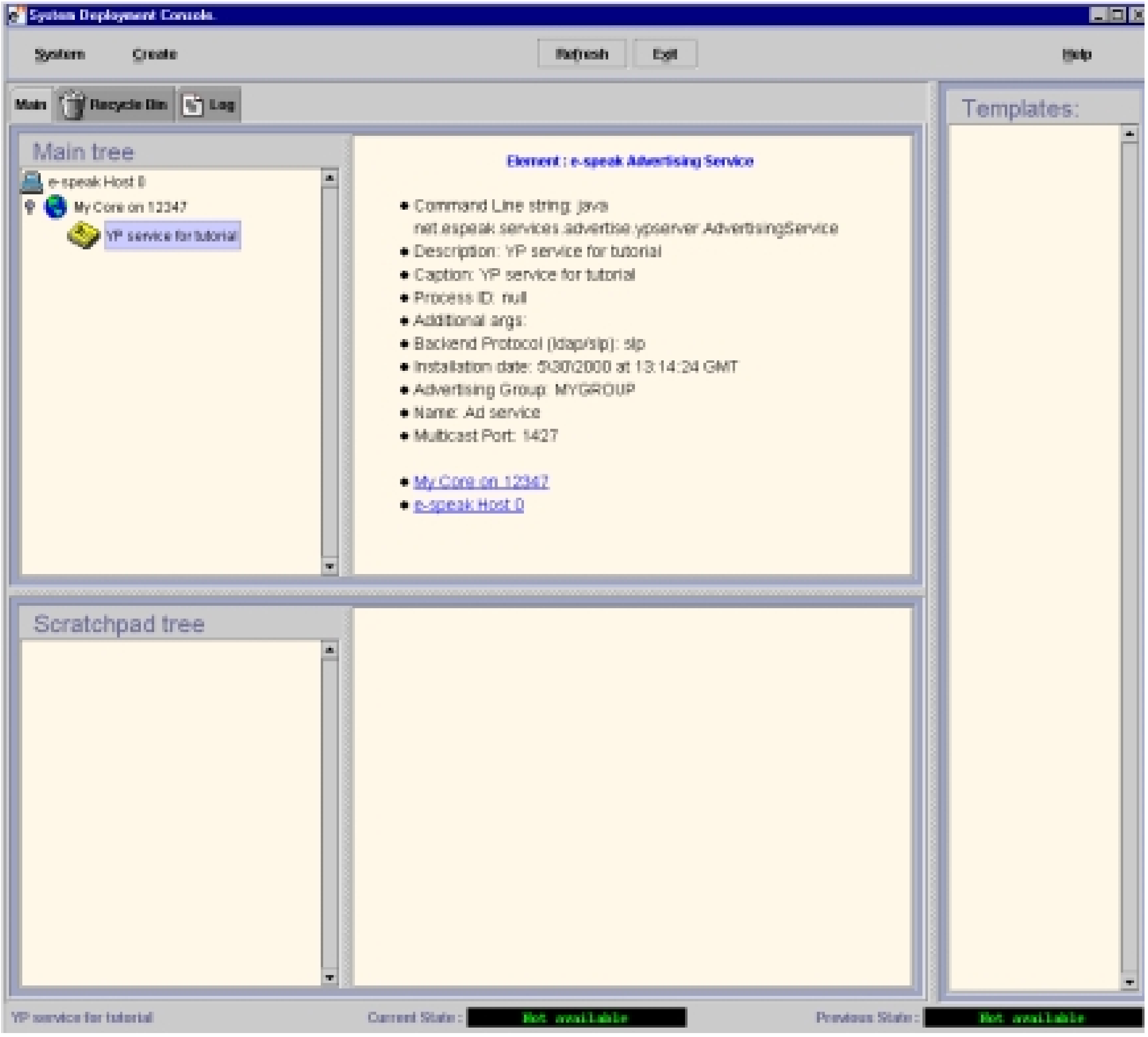


Configuring YP service for tutorial

Description	YP service for tutorial
Caption	YP service for tutorial
Additional args	
Backend Protocol (ldap/slp)	slp
Advertising Group	MYGROUP
Name	Ad service
Multicast Port	1427

Figure 17 The property form for the Advertising Service

- Right click on the core and select ‘Associate with a hosted client...’
- In the window that pops up select the target client (the advertising service
- If the above steps were correct, the advertising service icon should dock under the core icon.
- Start the service.
- Figure 18 illustrates how the deployment should now appear in the Console.



Element : e-speak Advertising Service

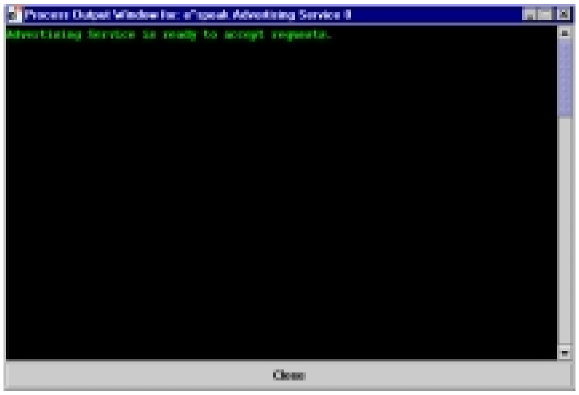
- Command Line string: java net.espeak.services.advertise.ypserver.AdvertisingService
- Description: YP service for tutorial
- Caption: YP service for tutorial
- Process ID: null
- Additional args:
- Backend Protocol (ldap/slp): slp
- Installation date: 5\30\2000 at 13:14:24 GMT
- Advertising Group: MYGROUP
- Name: Ad service
- Multicast Port: 1427

- My Core on 12347
- e-speak Host 0

YP service for tutorial Current State : Not available Previous State : Not available

Figure 18 The completed deployment after step four.

- Open the standard output window for the e-speak Advertising Service. The output should look as follows:

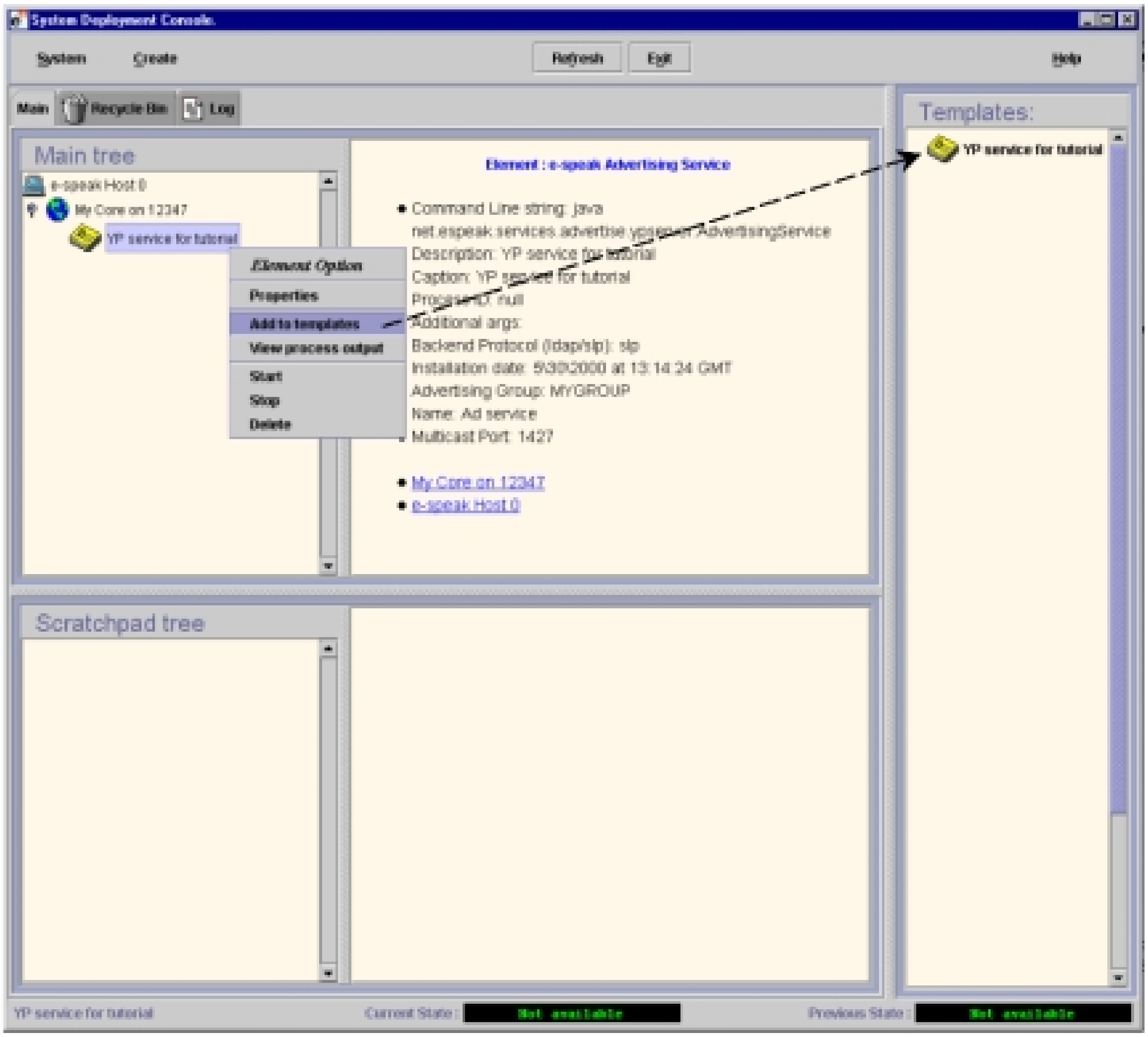


```
Advertising Service is ready to accept requests.
```

Figure 19 the output window for a correctly started advertising service.

Step four – Using templates

- To create a copy of the configured services we have just created we can use the Template Pane. This is useful if similar instances of these services are required in the future.
- Highlight the e-speak advertising service and from the popup menu select *'Add to templates'*.



Element : e-speak Advertising Service

- Command Line string: java net.espeak.services.advertise.ypserver.AdvertisingService
  Description: YP service for tutorial
  Caption: YP service for tutorial
  ProcessID: null
  Additional args:
  Backend Protocol (ldap/slp): slp
  Installation date: 5/30/2000 at 13:14:24 GMT
  Advertising Group: MYGROUP
  Name: Ad service
- Multicast Port: 1427

- My Core on 12347
- e-speak Host 0

Figure 20 the process of adding an element to the template pane.

- Next, permanently delete the services either by moving it to the recycle bin then emptying the bin, or by pressing <CRTL-Delete> when the service is selected.
- To recreate the original configuration you can now use the element from the template. Select each of the icons in the Template Pane and choose *'Add'* from the popup window.
- Create associations between the advertising service and the core.

- Start the service and check the output window for any problems.

Saving and Loading templates

- To save the templates to a file simply choose the *'Save template to file'* option from the *'System'* menu on the main menu bar. You will be presented with a file chooser and asked to select a file to save to templates to.
- To load the templates from a file, close the Console. Re-open the Console and choose the *'Load template'* option from the *'System'* menu on the main menu bar. Using the file chooser select the file you saved the data to. The templates should load back into the Template Selector Pane.

Step five – Starting a non-e-speak task

- Non-e-speak processes can be started in much the same way as e-speak cores are started. These native processes can be started directly under hosts (in the same way that cores can be started).
- Start a native task such as a directory listing command on the connected host by right-clicking on the host a selecting *'Create a…/User defined task'*. Fill in the task configuration form with the command line field reading 'dir' (Dos), or 'ls' (Unix). After starting the task, view the output in the process output window (<*CTRL-O*>, or from the popup menu).

Step six – Starting a user-defined e-speak client

- Though the Console has many useful e-speak services built-in, it may be necessary to start services that are not provided in the current schema (see Database Concepts). This can easily be done by requesting the Console to start a 'User Defined e-speak Client'. Try right-clicking on the core in your current deployment and selecting to start such a task. In the command line field of the configuration form that pops up enter the following:

```
java net.espeak.services.management.logger.manager.ESLogServiceManager localhost 12347
```

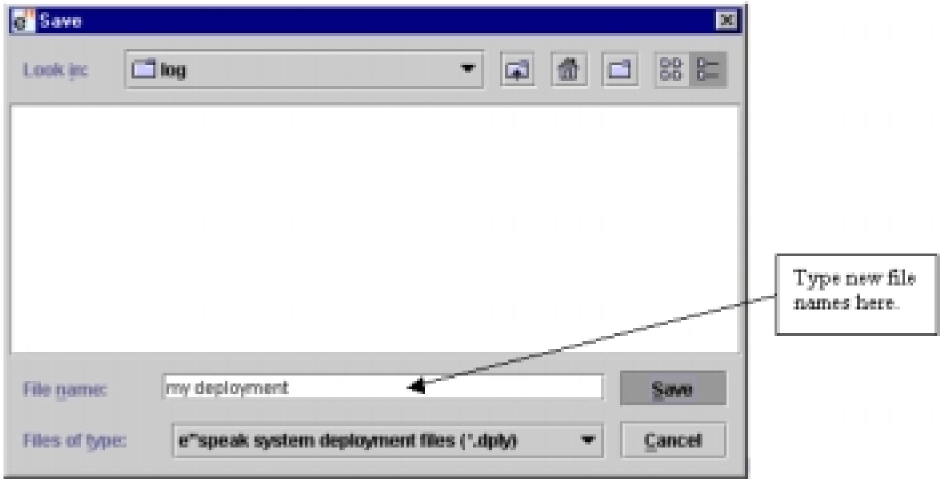
- Start the service and look at its output.
- You can start this service in the conventional way by selecting from the core's popup menu up to *'Create a…e-speak logger service'.* If you do this and start the service, you should find the output to be identical to the hand-configured service.

Step seven- Remote saving of deployments

- The e-speak enabled host will automatically save all the deployment data that you create whilst using the Console. Loading up the deployment created into this tutorial, simply requires you to re-connect to the host from Step one – Starting the Console. Once connected, you will see processes that were in either the Scratchpad tree or the Main tree reappear as before.

Step eight – Local saving of reusable deployments/scripts

- Deployments can be saved locally to a file. These deployments are reusable and can be instantiated on any number of hosts. Right-click on your host to save the current configuration. Choose the *'Save deployment to file'* option. Use the file chooser to pick a directory. In the **File name** text area you can type the file name for your deployment file. The file will be saved with the .dply extension. Now permanently delete all the processes under your host.



Look in: log

File name: my deployment

Files of type: e"speak system deployment files (*.dply)

Type new file names here.

Figure 21 The file chooser window

Step nine– Loading saved deployments/scripts

- The deployment saved in step eight can be reloaded onto your host. To do this right click on the host and choose the *'Load deployment from file'* option. Open the file that you saved the deployment in. The Elements will reappear under the host. These Elements require restarting.

Glossary of terms

Association – a model use to represent relation between elements.

Association Creator - the component in the Console that allows a user to interactively and explicitly define associations between tasks.

CIM (*Common Information Model)* – the distributed management model that Console is based on.

Console – The System Deployment Console.

Daemon Core - a specialized core running some essential management services.

Element – an object managed by the Console

E-speak Enabled Host – a host machine running a Daemon Core.

Log – a log of all user interactions with the Console.

Main Tree - the main tree containing all fully configured and manageable tasks.

Metadata – the meta-information about tasks used by Console to controls how it manages a task, its properties, and its associations.

Permanent Deletion – A 'hard' delete that stops a task then removes it from the database bypassing the Recycle Bin.

Process-level Management – managing tasks at the process level; being able to stop, start, and monitor processes.

Process Output Window - the component in the Console that displays the *Standard Output.*

Property – attribute of elements. Each element can be zero to many properties.

Property Form – a component used to configure a task or connection interactively.

Recycle Bin – a temporary, non-persistent store for tasks removed from the management interface.

Restore – the process of moving a task from the Recycle Bin back into the main database, thus re-attaching it to the management interface.

Schema – the set of metadata for every task-type and association-type recognised by Console.

Scratchpad Tree – the bottom left component of the Console. Used to display tasks that are not fully configured, in order to maintain a clear hierarchy and structure in the Main tree.

Standard Output - the textual output from a process.

Status Bar – the bar along the bottom of the Console window indicating the state of the selected task.

System Deployment – The name given to the configuration of tasks running on host machines.

Task – a logical process.

Template Pane - a pane on the right hand side of the Console that displays Templates.

Templates - useful, partially configured tasks. Can be saved and read from a file. Suitable for persistently storing customized configurations.

User Defined E-speak Client – an undefined e-speak client that can be defined interactively by the user.

User Defined Task - a non-e-speak task that can be defined interactively by the user.

Table of figures

Index