

# **Distributed Service Handler (DiSH)**

## **Administrator's Guide**

Information in this document is subject to change without notice. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express permission of Autonomy Systems Ltd.

Windows is a trademark of Microsoft Corp.

Unix is a trademark of X/Open Ltd.

Copyright © 2001 Autonomy Systems Ltd.

DiSH is a trademark of Autonomy Systems Ltd.

# Table of Contents

<b>1. Introduction.....</b>	<b>1</b>
System architecture.....	2
<b>2. Installing DiSH.....</b>	<b>3</b>
System requirements.....	3
Installing under Windows.....	4
Installing under UNIX.....	6
<b>3. Configuring DiSH.....</b>	<b>9</b>
[License] section.....	9
[Service] section.....	10
[Default] section.....	11
[ChildServices] section.....	12
[ExampleChild] section.....	13
[Schedule] section.....	14
[ExampleScheduleAction] sections.....	15
Example [ScheduleRestart] section.....	16
Example [ScheduleCheckStatistics] section.....	18
Example [ScheduleKeepAlive] section.....	20
Example [ScheduleAuditSearchFetch] section.....	23
Example DiSH configuration file.....	25
Adding a new child to the DiSH configuration file.....	26
<b>Appendix A: Service settings.....</b>	<b>27</b>
<b>Appendix B: Making HTTP requests to DiSH and its children.....</b>	<b>28</b>
<b>Appendix C: Standard Status and Control commands.....</b>	<b>29</b>
<b>Appendix D: Child actions.....</b>	<b>30</b>
<b>Appendix E: DiSH actions.....</b>	<b>31</b>
<b>Glossary.....</b>	<b>32</b>
<b>Index.....</b>	<b>33</b>

## Autonomy

---

Autonomy's infrastructure technology enables computers to form an understanding of text, Web pages, e-mails, voice, documents and people.

Because of this unique ability, Autonomy's technology automates business operations on unstructured information, such as categorizing, linking, personalizing and delivering operations, which, to date, have been completely manual.

Autonomy's technology powers any application dependant upon unstructured information including: e-commerce, customer relationship management, knowledge management, enterprise information portals and online publishing.

Autonomy was founded in 1996 and has offices in Boston, Chicago, Dallas, San Francisco, New York, and Washington, D.C. in the United States, as well as offices through Europe, including Amsterdam, Brussels, Cambridge, Frankfurt, Milan, Paris, Oslo, and Sydney. In July 1998, the company went public on the EASDAQ exchange (EASDAQ:AUTN). Autonomy floated on The Nasdaq National Market (NASDAQ: AUTN) in May 2000, and on the London Stock Exchange (LSE: AU.) in November 2000.



To contact Autonomy, please get in touch with your nearest location listed below.

### **Europe and South Pacific**

Autonomy Systems Ltd.  
Cambridge Business Park  
Cowley Road  
Cambridge  
CB4 0WZ

Help Desk: +44 (0) 800 0 282 858  
Switchboard: +44 (0) 1223 448 000  
Fax: +44 (0) 1223 448 001  
E-mail for information: [autonomy@autonomy.com](mailto:autonomy@autonomy.com)  
for support: [uksupport@autonomy.com](mailto:uksupport@autonomy.com)

The Help Desk operates from 9.30 am to 6.00 pm (GMT) Monday to Friday.  
Website: [www.autonomy.com](http://www.autonomy.com)

### **USA**

Autonomy Inc.  
301 Howard Street  
22<sup>nd</sup> Floor  
San Francisco  
CA 94105

Help Desk: +1 877 333 7744  
Switchboard: +1 415 243 9955  
Fax: +1 415 243 9984  
E-mail for information: [info@us.autonomy.com](mailto:info@us.autonomy.com)  
for support: [support@us.autonomy.com](mailto:support@us.autonomy.com)

The Help Desk operates from 9.30 am to 6.00 pm (CST) Monday to Friday, toll-free.  
Website: [www.autonomy.com](http://www.autonomy.com)



# Welcome

---

Thank you for choosing Autonomy and welcome to your DiSH™ version 2.2 Administrator's Guide.

## Autonomy Solutions

Autonomy solutions are built in a modular architecture, based on the Autonomy Application Builder™ (API). They provide an infrastructure that is fully scalable and customizable according to customers' present and future needs. Autonomy solutions include:

- **Portal-in-a-Box™**, our comprehensive and fully-automated Information Portal for content-rich Internet and Intranet sites,
- **ActiveKnowledge™**, which conducts a real-time analysis of the ideas involved in the content of any opened application and provides real-time links to relevant internal and external information,
- **i-WAP™**, an add-on to Portal-in-a-Box™, automating the delivery and personalization of timely and relevant information to mobile users.
- **Autonomy Server™** is the core engine that provides a fully automated and precise means of retrieving information. It allows content to be queried in any language and any format, wherever it is stored, and presented with hyperlinks to similar information, automatically and in real-time.
- **Autonomy Update™** automatically creates a personalized report, informing individual users of developments that are relevant to their specific roles and interests. The alert capability enables users to keep track of late-breaking news, automatically and in real-time.
- **Autonomy Application Builder™** is a toolkit that enables companies and partners to customize Autonomy's products according to their individual requirements. The Application Programming Interfaces (APIs) have been modularized into functional suites to allow developers to plug in only those modules required for a given application. The APIs are distributed with the Autonomy Server™ and a set of sample code.

## Autonomy DiSH™

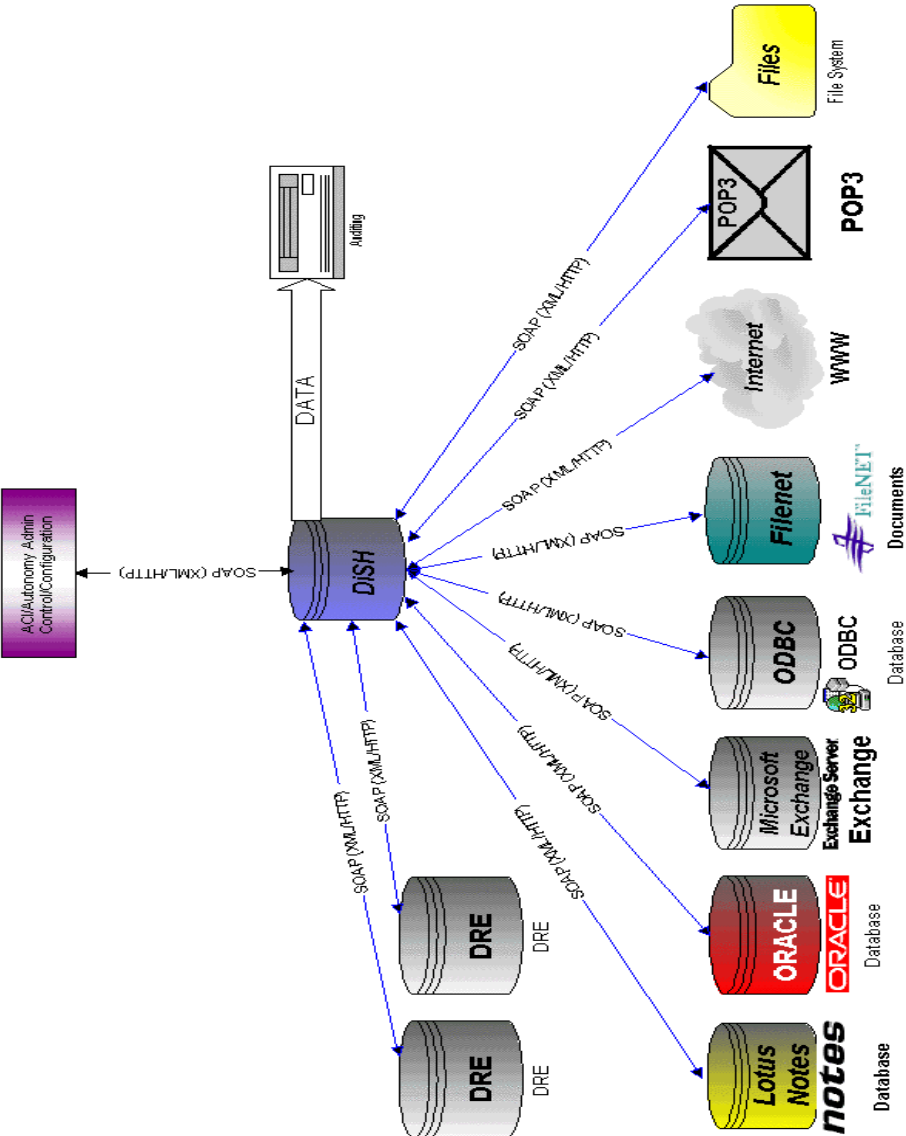
Autonomy's DiSH is the central point of control for all of the Autonomy services, such as DREs, Fetches, DIH or DQH. It allows you to control individual or multiple services simultaneously.

## 1. Introduction

---

Autonomy DiSH (Distributed Service Handler) serves as a middle layer to unify Autonomy with other enterprise systems and act as a META layer for controlling and monitoring all Autonomy applications. DiSH provides the crucial maintenance, administration, control and monitoring functionality of ACI, the Autonomy Content Infrastructure.

System architecture



## 2. Installing DiSH

---

### System requirements

**Platforms supported:**

- Microsoft Windows 98, NT 4 and 2000
- Solaris
- Linux

**Note:** other UNIX platforms can be supported on request.

**Minimum Server specification**

NT		UNIX	
200 MHz Pentium processor		128 MB of RAM	
64 MB RAM		1 GB hard disk recommended	
1 GB hard disk recommended			

## Installing under Windows

**Note:** DiSH should be installed by the system administrator.

To install under Windows insert the DiSH CD-ROM into your CD-Rom Drive.

If your Windows installation is configured to support it, inserting the CD-ROM will automatically start the DiSH installation program. Otherwise you can start the installation by double-clicking on the DiSH2.2.0.EXE program in the root directory of the CD-ROM through Windows explorer.

Read and follow all installation instructions on the screen carefully. Before the installation program can start to copy files onto your PC, you need to provide it with some information:

1. Read the license agreement and click on **Next** to accept it.
2. Check the license details are correct. Contact Autonomy Technical Support if they are incorrect or if you are missing the licensekey.txt file that contains your license key.  
If you are installing the DiSH 30 day evaluation version, enter the license holder.
3. Enter a unique name for the DiSH installation.  
**Note:** The unique name must not contain any spaces.
4. Select the directory in which you want to install DiSH. By default DiSH is installed on C:\Autonomy\DiSH, but you can use the **Browse** button to navigate to another location.
5. Enter the name of the program manager group to which you want to add the DiSH icons.
6. Enter the following:

### Service Port

The number of the port on which the DiSH listens for ACI commands.

### Service Control Clients

One or more IP addresses for machines that are permitted to control the DiSH. If you want to enter multiple addresses, you must separate the individual addresses with a comma (with no space). For example, **196.172.86.\*;196.172.87.\***

### Service Status Clients

One or more IP addresses for machines that are permitted to access information about the status of the service but are not permitted to control it. If you want to enter multiple addresses, you must separate the individual addresses with a comma (with no space). For example, **196.172.86.\*;196.172.87.\***

7. **Only for Windows NT and Windows 2000:** Specify whether you want to create a service for the DiSH. If you do, specify whether you want to start it immediately.

**Note:** it is useful to create services as you will be able to control the running of the DiSH using Windows Services.

8. Specify whether you want to add shortcuts to the Windows Start menu and click on **Next**.
9. You are now ready to specify in the DiSH configuration file which services you want to control via DiSH. Click on **Finish** to complete the installation of DiSH.

## Installing under UNIX

**Note:** DiSH should be installed by the system administrator.

To install under Unix insert the DiSH CD-ROM into your CD-Rom Drive. Read and follow all installation instructions on the screen carefully.

1. Copy the DiSH installer from the CD to your local disk.

2. Uncompress the installer using the command:

```
uncompress <Installer>.tar.z
```

3. Un-tar the resulting file using the command:

```
tar -xvf <Installer>.tar
```

This extracts the following files into the current working directory:

```
Setup.sh  
dish.cfg  
dish.exe  
dishfiles.txt  
freplace.exe  
startdish.sh  
uninstall.sh
```

4. You next need to run the installer script, `Setup.sh` with the following arguments:

**<Base\_Name>**

Name of the instance of the DiSH

**<Main\_dir>**

Full path of the main installation directory

**<Service Port for Distributed Service Handler>**

The number of the port on which the DiSH listens for ACI commands.

**<Service Control Client CSV List>**

One or more IP addresses for machines that are permitted to control the DiSH. If you want to enter multiple addresses, you must separate the individual addresses with a comma (with no space). For example, **196.172.86.\*,196.172.87.\***

**<Service Status Client CSV List>**

One or more IP addresses for machines that are permitted to access information about the status of the service but are not permitted to control it.

If you want to enter multiple addresses, you must separate the individual addresses with a comma (with no space). For example, **196.172.86.\*,196.172.87.\***

If you want to grant access to any machine, you can enter an asterix in quotes ("**\***").

5. The DiSH installation ins now complete and you are ready to specify in the DiSH configuration file which services you want to control via DiSH.



### 3. Configuring DiSH

---

The number of sections that the DiSH configuration file contains varies according to the number of children that you want to control through DiSH and the number of scheduled actions that you have specified.

The DiSH configuration file can contain the following sections:

- [License]
- [Service]
- [Default]
- [ChildServices]
- [<Name\_of\_Child>]
- [Schedule]
- [<Name\_of\_Schedule\_item>]

**Note:** the following settings are interchangeable

- TRUE = ON = Y = 1
- FALSE = OFF = N = 0

#### [License] section

This section contains the licensing details. You should not edit this section, as this could stop the DiSH functioning.

**Key**

The license key.

**Holder**

The name of the license holder.

## Configuring DiSH

### **[Service] section**

This section determines which machines are permitted to use and control the DiSH service.

#### **ServicePort**

DiSH listens for commands on this port

#### **ServiceStatusClients**

One or more IP addresses for machines that are permitted to access information about the status of the service but are not permitted to control it.

If you want to enter multiple addresses, you must separate the individual addresses with a comma (with no space).

For example, **196.172.86.\*;196.172.87.\***

#### **ServiceControlClients**

One or more IP addresses for machines that are permitted to control the service.

If you want to enter multiple addresses, you must separate the individual addresses with a comma (with no space).

For example, **196.172.86.\*;196.172.87.\***

## [Default] section

DiSH uses the values that you set in the [Default] section for all children by default. If you want to specify different settings for an individual child, you can set them in the [ExampleChild] section, in which case the default settings for this child will be overridden.

### LogFile

This is the default log file for all children. By default this is **Child.log**.

### NumericDates

Specifies the format of the date that is used for auditing:

#### True

If **NumericDates** is set to **True**, any date in the audit log files is in numeric format (that is the number of seconds since 1970). For example, if a log entry was made on Friday, the 23<sup>rd</sup> of April 1999 at 06:05:22am, the log entry would be **9514778514**. This is the default setting.

#### False

If **NumericDates** is set to **False**, any date in the audit log files is in date format. For example, **Wed Aug 09 17:37:26 2000**.

### SMTPServer

The IP address or the host name of your mail server.

### SMTPPORT

The number of the port that your mail server uses for SMTP.

### ChildControlMethod

Allows you to specify how children should be controlled:

#### HTTP

The children are controlled via HTTP requests. This is the default.

#### SCM

The children are controlled via the Service Control Manager

## [ChildServices] section

<Number>=<n>

<Number>

The number of child sections. By default this is **2**.

<n>

The name of the child section. You can list all the child sections that you want to control. For example:

0=<Name\_of\_ChildSection1>

1=<Name\_of\_ChildSection2>

### Note:

- the list of childsections must start at 0, and the number of childsections that you list must correspond to the number of children that you have specified.
- Any childsection that you are listing must refer to an actual **[ExampleChild] section** in the DiSH configuration file.

## [ExampleChild] section

This section contains the details of the first child action controlled by DiSH. This action is therefore not necessarily called [ExampleChild] but has the name of your first child action.

### ChildHost

The IP address of the machine that hosts the child.

**Note:** If you want to start and stop a child, the child must be installed on the same machine as the DiSH.

### ChildPort

The port that the child uses for ACI communication.

### ChildControlMethod

Allows you to specify how children should be controlled:

#### HTTP

The children are controlled via HTTP requests. This is the default.

#### SCM

The children are controlled via the Service Control Manager

### ChildServiceName

The name of the service (for example, HTTPFetch, DQH and so on) that you want to control through DiSH. This should be the executable name minus the extension (provided an extension exists).

If you, for example, want to control HTTPFetch, enter:

```
ChildServiceName=<Your_HTTPFetch_installation_name>
```

### ChildPath

If **ChildControlMethod** is set to HTTP, you must specify a **ChildPath**.

Enter the path to the service (for example, HTTPFetch, DQH and so on) that you want to control through DiSH. This should be the path to the executable including the name of the executable.

If you, for example, want to control HTTPFetch using HTTP, enter:

```
ChildServiceName=<HTTPFetch_installation_location>/<HTTPFetch_installation_executable>
```

## [Schedule] section

The scheduler triggers child actions as if they were requests to DiSH. You can have multiple scheduled actions but each of the actions must have a distinct name. You can specify schedule details for each scheduled action.

**<Number>=<n>**

**<Number>**

The number of scheduled child sections.

**<n>**

The name of the scheduled child section. You can list all the scheduled child sections that you want to control. For example:

0=<Name\_of\_scheduled\_ChildSection1>

1=<Name\_of\_scheduled\_ChildSection2>

## [ExampleScheduleAction] sections

You can add this section to the configuration file in order to specify the details of the first action in the [Schedule] section.

### ChildAction

The child action that should be executed. You can enter one of the following:

#### GETSTATUS

Allows you to find out if a service is running.

#### GETSTATUSINFO

Allows you to find out the product name, version and so on.

#### GETSTATISTICS

Allows you to gather statistics from the service.

#### START

Allows you to start a service.

#### STOP

Allows you to stop a service.

#### PAUSE

Allows you to pause a service.

#### KEEPALIVE

Used as a scheduled action to restart a service every time it is stopped. You can apply it to more than one service.

#### RESTART

Restarts a service or services.

#### GETSYSTEMRESOURCES

Allows you to get information about system resources used by DiSH.

#### GETMEMORYSPACE

Allows you to get the memory space used by DiSH.

## Example [ScheduleRestart] section

### ScheduleInterval

You can set the interval in which the specified **ChildAction** is executed to. The default setting is **1DAY**.

#### Weeks

Enter the number of weeks that should elapse between the execution of any specified **ChildAction**. For example: **1 week**.

#### Days

Enter the number of days that should elapse between the execution of any specified **ChildAction**. For example: **1 day**.

#### Seconds

Enter the number of seconds that should elapse between the execution of any specified **ChildAction**. For example: **43200** to specify a 12 hour interval.

### TargetCSVs

The name of the child to which you want to apply the specified **ChildAction**. If you want to specify multiple children, you need to separate the individual children using commas (with no spaces).

### ChildAction

The child action that should be executed. You can enter one of the following:

#### GETSTATUS

Allows you to find out if a service is running.

#### GETSTATUSINFO

Allows you to find out the product name, version and so on.

#### GETSTATISTICS

Allows you to gather statistics from the service.

#### START

Allows you to start a service.

**STOP**

Allows you to stop a service.

**PAUSE**

Allows you to pause a service.

**KEEPALIVE**

Used as a scheduled action to restart a service every time it is stopped. You can apply it to more than one service.

**RESTART**

Restarts a service or services.

**GETSYSTEMRESOURCES**

Allows you to get information about system resources used by DiSH.

**GETMEMORYSPACE**

Allows you to get the memory space used by DiSH.

## Example [ScheduleCheckStatistics] section

This section sends you an e-mail if certain values don't reach or exceed a defined value.

### TargetCSVs

The name of the child to which you want to apply the specified **ChildAction**. If you want to specify multiple children, you need to separate the individual children using commas (with no spaces).

### EmailAlertField

A field in the child's configuration file from which DiSH reads a value. If the value is  $< =$  the value specified in EMAILALERTFIELDRANGEMIN or  $> =$  the value specified in EMAILALERTFIELDRANGEMAX, DiSH notifies you by sending you an e-mail.

### EmailAlertSection

The section in the child's results from which the value is read.

### EmailAlertFieldRangeMin

The minimum value that the EmailAlertField is permitted to have.

### EmailAlertFieldRangeMax

The maximum value that the EmailAlertField is permitted to have.

### ChildAction

The child action that should be executed. You can enter one of the following:

#### GETSTATUS

Allows you to find out if a service is running.

#### GETSTATUSINFO

Allows you to find out the product name, version and so on.

#### GETSTATISTICS

Allows you to gather statistics from the service.

#### START

Allows you to start a service.

**STOP**

Allows you to stop a service.

**PAUSE**

Allows you to pause a service.

**KEEPALIVE**

Used as a scheduled action to restart a service every time it is stopped. You can apply it to more than one service.

**RESTART**

Restarts a service or services.

**GETSYSTEMRESOURCES**

Allows you to get information about system resources used by DiSH.

**GETMEMORYSPACE**

Allows you to get the memory space used by DiSH.

**For example :**

```
[SCHEDULECHECKDREPERFORMANCE]
SCHEDULEINTERVAL=60
TARGETCSVS=ExampleChild
CHILD ACTION=GETSTATISTICS
CHILDRESPONSELOGFILE=ExampleChildAlerting.log
EMAILALERTFROMADDRESS=dish@autonomy.com
EMAILALERTTOADDRESS=myuser@autonomy.com
EMAILALERTSUBJECTLINE=DiSH_Alert
EMAILALERTFIELD=60SECONDREQUESTSPERSECOND
//the fieldname that you want to check
EMAILALERTSECTION=STATISTICS
//the section from which you want to retrieve the result
EMAILALERTFIELDRANGEMAX=10
//the maximum number of requests per 60 seconds, any larger will result in an e-mail
EMAILALERT=TRUE
```

## Example [ScheduleKeepAlive] section

This is a section of the configuration file, which specifies the details of the second of the scheduled actions specified in the [SCHEDULE] section above. This is an alerting feature, which will restart the children if they are stopped.

### ScheduleInterval

You can set the interval in which the specified **ChildAction** is executed to. The default setting is **1DAY**.

#### Weeks

Enter the number of weeks that should elapse between the execution of any specified **ChildAction**. For example: **1 week**.

#### Days

Enter the number of days that should elapse between the execution of any specified **ChildAction**. For example: **1 day**.

#### Seconds

Enter the number of seconds that should elapse between the execution of any specified **ChildAction**. For example: **43200** to specify a 12 hour interval.

### TargetCSVs

The name of the child to which you want to apply the specified **ChildAction**. If you want to specify multiple children, you need to separate the individual children using commas (with no spaces).

### ChildAction

The child action that should be executed. You can enter one of the following:

#### GETSTATUS

Allows you to find out if a service is running.

#### GETSTATUSINFO

Allows you to find out the product name, version and so on.

#### GETSTATISTICS

Allows you to gather statistics from the service.

**START**

Allows you to start a service.

**STOP**

Allows you to stop a service.

**PAUSE**

Allows you to pause a service.

**KEEPALIVE**

Used as a scheduled action to restart a service every time it is stopped. You can apply it to more than one service.

**RESTART**

Restarts a service or services.

**GETSYSTEMRESOURCES**

Allows you to get information about system resources used by DiSH.

**GETMEMORYSPACE**

Allows you to get the memory space used by DiSH.

**EmailAlert**

Enter **True** if an e-mail should be sent to **EmailAlertToAddress** or **False** if no e-mail should be send.

**EmailAlertFromAddress**

Enter the address that should be displayed as the sender of any e-mail that is sent to **EmailAlertToAddress**.

**EmailAlertFromHost**

The IP address or machine name of the host that should be displayed as the sender of any e-mail that is sent to **EmailAlertToAddress**.

**EmailAlertToAddress**

Enter the e-mail address of the person to whom e-mail should be sent.

**EmailAlertSubjectLine**

The subject of the email that is sent. The default is **DiSH Alert**.

## Configuring DiSH

### **EmailAlertSection**

The section in the child's results from which the value is read.

### **EmailAlertField**

The name of the field that DiSH should check to find out whether an alert is required. The default is **Response**.

### **EmailAlertFieldMustHaveCSVs**

If the **EmailAlertFieldMustHaveCSVs** value does not match the value contained in the field that is specified by **EmailAlertField** e-mail is sent.

### **EmailAlertFieldCantHaveCSVs**

If the **EmailAlertFieldCantHaveCSVs** value does match the value contained in the field that is specified by **EmailAlertField** e-mail is sent.

## Example [ScheduleAuditSearchFetch] section

This is a section of the configuration file that specifies the details of the third of the scheduled actions specified in the [SCHEDULE] section above. This is an auditing feature.

### ScheduleInterval

You can set the interval in which the specified **ChildAction** is executed to. The default setting is **1DAY**.

#### Weeks

Enter the number of weeks that should elapse between the execution of any specified **ChildAction**. For example: **1 week**.

#### Days

Enter the number of days that should elapse between the execution of any specified **ChildAction**. For example: **1 day**.

#### Seconds

Enter the number of seconds that should elapse between the execution of any specified **ChildAction**. For example: **43200** to specify a 12 hour interval.

### TargetCSVs

The name of the child to which you want to apply the specified **ChildAction**. If you want to specify multiple children, you need to separate the individual children using commas (with no spaces).

### ChildAction

The child action that should be executed. You can enter one of the following:

#### GETSTATUS

Allows you to find out if a service is running.

#### GETSTATUSINFO

Allows you to find out the product name, version and so on.

#### GETSTATISTICS

Allows you to gather statistics from the service.

#### START

Allows you to start a service.

## Configuring DiSH

### **STOP**

Allows you to stop a service.

### **PAUSE**

Allows you to pause a service.

### **KEEPALIVE**

Used as a scheduled action to restart a service every time it is stopped. You can apply it to more than one service.

### **RESTART**

Restarts a service or services.

### **GETSYSTEMRESOURCES**

Allows you to get information about system resources used by DiSH.

### **GETMEMORYSPACE**

Allows you to get the memory space used by DiSH.

## **FieldCSV**

Enter the names of the fields that you want DiSH to log in the Audit log. If you enter multiple field names, separate them with commas (with no spaces).

If several fields have the same name, all fields that have this name will be printed out. In this case the fields are printed in the order in which they appear.

## **ChildResponseLogFile**

Enter the name of the log file in which DiSH should log the results of the **ChildAction**.

## Example DiSH configuration file

The following is an example of a DiSH configuration file:

The number of sections in your configuration file depends on the number of child actions and scheduled actions that you have.

```
[License]
Holder=Autonomy
Key=xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

[Service]
SERVICEPORT=12345
SERVICESTATUSCLIENTS=*
SERVICECONTROLCLIENTS=*

[DEFAULT]
CHILDHOST=127.0.0.1
CHILDCONTROLMETHOD=HTTP
SMTPSERVER=uk.mycompany.com
SMTPPORT=25
LOGFILE=DiSHChildren.log

[ChildServices]
Number=1
0=ExampleChild

[ExampleChild]
CHILDPORT=10001
CHILDSERVICENAME=ExampleChild
CHILDPATH=ExampleChild.exe
```

## Configuring DiSH

```
[Schedule]
Number=1
0=ExampleScheduleAction

[ExampleScheduleAction]
SCHEDULEINTERVAL=600seconds
TARGETCSVS=ExampleChild
CHILDACTION=GETSTATISTICS
CHILDRESPONSELOGFILE=ExampleChildAudit.csv
```

## Adding a new child to the DiSH configuration file

### To add a new child to DiSH:

1. Add a new section to the DiSH configuration file.
2. Add a key to the [ChildServices] section that points to this new section.

### For example:

```
[ChildServices]
Number=2
0=ExampleChild
1=MyNewHTTPFetch

[MyNewHTTPFetch]
CHILDPORT=1523
CHILDSERVICENAME=MyNewHTTPFetch
CHILDPATH= /Autonomy /HTTPFetch /MyNewHTTPFetch.exe
```

You should then restart the DiSH.

## Appendix A: Service settings

DiSH behaves as a standard Autonomy Service. If the following settings are added to the DiSH configuration file (or the configuration file of another Autonomy ACI product), the service port is enabled and will accept the standard status and control commands.

[Service]	Description	Default
ServicePort	The number of the port on which the DiSH listens for ACI commands.	None
ServiceControlClients	One or more IP addresses for machines that are permitted to control the DiSH. If you want to enter multiple addresses, you must separate the individual addresses with a comma (with no space). For example, <b>196.172.86.*;196.172.87.*</b>	None
ServiceStatusClients	One or more IP addresses for machines that are permitted to access information about the status of the service but are not permitted to control it. If you want to enter multiple addresses, you must separate the individual addresses with a comma (with no space). For example, <b>196.172.86.*;196.172.87.*</b>	None

## Appendix B: Making HTTP requests to DiSH and its children

---

You can submit HTTP requests to Dish in the following ways.

**Note:** to stop and start services you can use the NT Service Control Manager instead of HTTP requests.

### Submitting HTTP request directly to DiSH

You can submit HTTP requests directly to DiSH (or another ACI compatible Autonomy product) by using the ACI port defined in the configuration file's [Service] section of the Autonomy product that you are controlling.

Enter http requests in the following form:

**http ://<ipaddress>:<ACI\_Service\_Port>/ACTION=<An\_Action>**

For example:

**http ://10.9.8.37:30000/ACTION=GETSTATISTICS**

### Submitting HTTP request via DiSH to its children

You can submit HTTP requests via DiSH (or another ACI compatible Autonomy product) to its children by using the ACI port defined in the [Service] section of the DiSH configuration file.

Enter http requests in the following form:

**http ://<ipaddress>:<ACI\_Service\_Port>/ACTION=CHILD ACTION&CHILD ACTION=<An\_Acti on>&TargetCSVs=<ChildName>**

**Note:** You can enter multiple targets by using wildcards or a comma separated list of <childname> (with no spaces).

**For example:**

**http ://<ipaddress>:<ACI\_Service\_Port>/ACTION=CHILD ACTION&CHILD ACTION=<An\_Acti on>&TargetCSVs=\*fetch\***

In this example the command is sent to all children whose name contains the word "fetch".

**http ://<ipaddress>:<ACI\_Service\_Port>/ACTION=CHILD ACTION&CHILD ACTION=<An\_Acti on>&TargetCSVs=MyFetch,MyDRE**

Appendix C: Standard Status and Control commands

In this example the command is sent to the children **MyFetch** and **MyDRE**.

## Appendix C: Standard Status and Control commands

You can use the following commands to control any Autonomy ACI enabled product.

ACTION	Control/ Status	DESCRIPTION
GETAUDIT	STATUS	This returns the location of the audit file
GETSTATUS	STATUS	This is used to find out if a service is running
GETSTATUSINFO	STATUS	This is used to find out the product name, version etc.
GETCONFIG	STATUS	This is used to obtain the configuration file
SETCONFIG	CONTROL	This is used to set the configuration file
GETSTATISTICS	STATUS	This is used to gather statistics from the service
START	CONTROL	This is used to start a service
STOP	CONTROL	This is used to stop a service
PAUSE	CONTROL	This is used to pause a service
MERGECONFIG	CONTROL	Merge the fragment of the config file passed with the services config file.

# Appendix D: Child actions

The following actions can be sent to any Autonomy ACI enabled product by using an HTTP request.

ACTION	DESCRIPTION
KEEPALIVE	This is used to restart a service every time it is stopped. It is used as a scheduled action. Can be applied to more than one service.
RESTART	This is used to restart a service or services

## Appendix E: DiSH actions

---

The following actions can be sent to DiSH by using an HTTP request. They cannot be sent to any other Autonomy ACI enabled product.

ACTION	DESCRIPTION
GETSYSTEMRESOURCES	Used to get information about system resources used by DiSH
GETMEMORYSPACE	Used to get the memory space used by DiSH

---

## Glossary

### ACI

Autonomy Content Infrastructure.

### Autonomy Search

Unlike ordinary searches that look for keywords the Autonomy Search allows you to enter a natural language query. The concept of the query is analyzed and documents that are relevant to this concept are returned to you.

### Database

An Autonomy database is a data pool that is contained within the **DRE**. You can retrieve information that has been indexed into the **DRE** from the database, for example, through submitting a query to the **DRE**.

### DRE (Dynamic Reasoning Engine)

Commerce Application Builder contains a Products DRE, a Category DRE and a Orders DRE. Each DRE contains a **database** that stores information. A customer or the administrator can retrieve information from a DRE, for example, by conducting a search.

Each DRE contains a **database** that stores information. You can retrieve information from a DRE, for example, by conducting a search.

### Indexing

Documents are converted to an .idx file and then loaded into a DRE. This process is called "indexing".

## Index

---

### A

- ACL, 1, 4, 6, 13, 27, 28, 29, 30, 31, 32
- Architecture, 2
- Autonomy Search, 32

### C

- Child actions, 30
- [ChildServices] section, 12
- Commands
  - Control, 29
  - Status, 29
- Configuring DiSH, 9

### D

- Database, 32
- [Default] section, 11
- DiSH
  - Actions, 31
  - Configuration file, 9
  - Configuring, 9
  - Installing, 3
  - Settings, 9
  - System architecture, 2
- DiSH configuration file
  - Adding a new child, 26
  - Example, 25

### E

- [ExampleChild] section, 13
- Example DiSH configuration file, 25
- [ExampleScheduleAction] sections, 15

### H

- HTTP requests, 28

### I

- Indexing, 32
- Installation, 3
- Installing DiSH
  - System requirements, 3
  - Under UNIX, 6
  - Under Windows, 4
- Interchangeable settings, 9

### L

- [License] section, 9

### P

- Platforms supported, 3

### S

- [Schedule] section, 14
- Section
  - [ChildServices], 12
  - [Default], 11
  - [ExampleChild], 13
  - [License], 9
  - [Schedule], 14
  - [Service], 10
- Sections
  - [ExampleScheduleAction], 15
- Server specification, 3
- Service
  - Control Clients, 4, 27
  - Port, 4, 27
  - Status Clients, 4, 27
- [Service] section, 10
- Service settings, 27
- Status commands, 29
- Supported platforms, 3
- System architecture, 2
- System requirements, 3