



# **ACP portal**

## **Administrator manual**

### ***Release 3.1***

**Author**

Ben van Balen

**Date**

01-04-2011

**Document version**

3.1

**Status**

Draft



## Versions

Version history			
Version	Date	Changes	Author
0.1	15-12-2007	New Administrator manual created from the "ACP technical description v1.5" document	Ben van Balen
1.0	21-12-2007	Definitive version for ACP v1.2	Ben van Balen
1.3	11-04-2008	Update for ACP version 1.3	Ben van Balen
2.0	29-09-2008	Update for ACP version 2.0	Ben van Balen
2.1	01-04-2009	Update for ACP version 2.1	Ben van Balen
2.2	01-10-2009	Update for ACP version 2.2	Ben van Balen
2.3	01-04-2010	Update for ACP version 2.3	Ben van Balen
3.0	01-10-2010	Update for ACP version 3.0	Ben van Balen
3.1	01-04-2011	Update for ACP version 3.1	Ben van Balen

Distribution		
Version	Date	Name
0.1	11-12-2007	Richard Gillingham, Niels van der Waart, Vijay Viswanathan, Luck Prins
1.0	21-12-2007	Richard Gillingham, Niels van der Waart, Vijay Viswanathan, Luck Prins, Hemanth Kumar C, Sven-Erik de Weerd
1.3	11-04-2008	Richard Gillingham, Niels van der Waart, Vijay Viswanathan, Luck Prins, Hemanth Kumar C, Sven-Erik de Weerd
2.0	01-10-2008	Tools & Automation workspace
2.1	01-04-2009	Tools & Automation workspace
2.2	01-10-2009	Tools & Automation workspace
2.3	01-04-2010	Tools & Automation workspace
3.0	01-10-2010	Tools & Automation workspace; <a href="http://goto/acp">http://goto/acp</a> ; with the ACP application
3.1	01-04-2011	Tools & Automation workspace; <a href="http://goto/acp">http://goto/acp</a> ; with the ACP application

Review			
Version	Date	Name	Approved
0.1	11-12-2007	Richard Gillingham, Niels van der Waart, Vijay Viswanathan, Luck Prins	14-12-2007
2.0	29-09-2008	Richard Gillingham, Niels van der Waart, Vijay Viswanathan, Luck Prins, Saroja Kanta Bal, Hemanth Kumar C	
2.1	25-03-2009	Richard Gillingham, Niels van der Waart, Vijay Viswanathan, Luck Prins, Saroja Kanta Bal	30-03-2009
3.0	13-08-2010	ETA ACP Support	
3.1	17-03-2011	ETA ACP Support; Paul Speyer; Matthew Willsher; Okko Jager	



# Contents

VERSIONS	1
INTRODUCTION	4
1 ACP TECHNOLOGY	5
1.1 USED SOFTWARE	5
1.2 RRDTOOL INTERNALS	6
2 ACP ARCHITECTURE	8
2.1 LOCAL STORAGE INFRASTRUCTURE	8
2.2 SHARED STORAGE INFRASTRUCTURE	8
2.3 MYSQL DATABASE	9
2.4 STARTING AND STOPPING ACP	12
2.5 BACKUP AND RESTORE	12
3 ADDITIONAL CONFIGURATION	13
3.1 ACP CONFIGURATION FILE	13
3.2 DNS	14
3.3 EMAIL	15
3.4 TIME ZONE	16
3.5 NETWORK TIME PROTOCOL	16
4 FETCHERS	17
4.1 FETCHERS AND CONFIG FILES	17
4.2 LOGGING	18
4.3 ACP RRD FILES	18
4.4 SCHEDULING	19
4.5 ACP MAINTENANCE JOBS	19
5 OBJECT TYPES	20
5.1 SUPPORTED OBJECT TYPES	20
5.2 HOW TO CONFIGURE OBJECTS FOR ACP MONITORING	20
6 ADD & MODIFY OBJECTS	41
6.1 ADDING A CUSTOMER	41
6.2 ADDING AN OBJECT	41
6.3 MODIFYING OR DELETING AN OBJECT	43
7 TROUBLESHOOTING CONNECTIVITY	44
7.1 MANUAL FETCHER RUNS	44
7.2 GENERIC TESTS	45
7.3 SNMP	45
7.4 SSH	46
7.5 HTTP(S)	46
7.6 ORACLE	46
7.7 SQL SERVER	47
7.8 MYSQL	47
7.9 WINDOWSWMI	48



8	MANAGING INFRA CHAINS	49
8.1	LAYERS	49
8.2	TIERS	50
8.3	ADDING OR MODIFYING AN INFRA CHAIN	51
9	TREND BREACH DETECTION AND ALERTING	52
9.1	INTRODUCTION	52
9.2	HOLT-WINTERS ABERRANT BEHAVIOR DETECTION	52
9.3	SETTING PORTAL WIDE DEFAULTS	57
9.4	ADDING AN OBJECT	58
9.5	MODIFYING AN OBJECT	59
9.6	THE HOLT-WINTERS PARAMETER FILE FOR AN OBJECT	59
9.7	CHANGING HOLT-WINTERS PARAMETERS OF AN EXISTING RRD FILE	60
9.8	TREND BREACH ALERT SETTINGS	62
9.9	VIEWING TREND BREACH ALERTS	63
10	ADMINISTRATION MENU	66
10.1	PORTAL SETTINGS	66
10.2	CONFIG FILES	72
10.3	LOG FILES	75
10.4	RRD FILES	76
10.5	AVAILABILITY MANAGEMENT	76
10.6	USER ADMINISTRATION	77
10.7	MENU ACTIONS	78
10.8	TOOLS	78

## List of figures

Figure 1	Shared storage ACP infrastructure in NL	8
Figure 2	Creating an Oracle client-server connection	31
Figure 3	The add object screen for an Oracle database	42
Figure 4	The modify object screen of an Oracle database	43
Figure 5	Layers in the infrastructure	49
Figure 6	Example of an infra chain	50
Figure 7	The Modify Infra Chain screen	51
Figure 8	Trend breach examples	55
Figure 9	The Holt-Winters settings screen	57
Figure 10	Change Holt-Winters parameters in an RRD file	60
Figure 11	Trend breach alert settings	62
Figure 12	Outstanding trend breach alerts	63
Figure 13	Trend breach alert history	64
Figure 14	Contents of the Multicenter logfile	65
Figure 15	The Global settings screen	66
Figure 16	The Fetcher nodes screen	67
Figure 17	The Object settings screen	68
Figure 18	The Icon settings screen	69
Figure 19	The Contract settings screen	70
Figure 20	Service window definition for non-24x7 contracts	71



# Introduction

The Availability, Capacity and Performance portal (ACP portal) is an internally developed tool for capacity management, trending, analysis and reporting for the entire ITIL infrastructure. It is designated as the primary tool for availability, capacity and performance monitoring and reporting in the global service catalog.

The goal of ACP portal is to record a metrics history of every managed object and to generate graphs from this history for any user-specified period. This document serves as a short manual for ACP portal version 3.1 super users and administrators. The information in this document is complementary to the ACP portal User Manual.

April 2011  
Outsourcing Services NL

Ben van Balen  
Product Support and Development Manager



# 1 ACP technology

## 1.1 Used software

ACP's functioning relies on a number of open source software components. The ACP code itself however, is not open source software and is only to be distributed to Logica staff.

ACP version 3.1 uses the following software components:

- Debian Linux 5.0 (Lenny) release 8
- Apache 2.2.9 (deb package) (<http://httpd.apache.org>)
- PHP 5.2.6 (deb package) (<http://www.php.net>)  
Programming language used for the web interface.
- dhtmlXtree (<http://www.dhtmlx.com/docs/products/dhtmlxTree>)  
Used for displaying managed objects in an AJAX based tree menu.
- MySQL Community Server 5.0.51a (deb package) (<http://www.mysql.com>)  
Used for storing the data of the tree menu and additional data used for ACP.
- RRDtool 1.3.1 (deb package) (<http://www.rrdtool.org>)  
Used for storing collected data and generating graphs.
- HTMLDoc 1.8.27 (deb package) (<http://www.htmldoc.org>)  
Used for generating PDFs from ACP's HTML pages.
- Hping2 2.rc3 (deb package) (<http://www.hping.org>)  
Used to ping objects (requires root privileges)
- Oracle client software 9.2.0.4 (precompiled software tree) (<http://otn.oracle.com>)  
Used for the Oracle fetcher.
- Python 2.5.2 (deb package) (<http://www.python.org>)  
Used for fetcher scripts
- FreeTDS 0.82 (deb package) (<http://www.freetds.org>)  
Open source software for SQL Server connections from Linux. Used for the SQL Server fetcher.
- Net-SNMP 5.4.1 (deb package) (<http://www.net-snmp.org>)  
Used for executing snmp requests. Used in the following fetchers: Windows, Unix generic fetcher, all network devices, Netapp filer, VMWare.
- Anfy text scroll applet (<http://www.anfyteam.com>)  
Used for the About screen.
- Subversion (<http://subversion.tigris.org>)  
Used for version control
- Sudo 1.6. (<http://www.courtesan.com/sudo>)
- OpenSSH 1:5.1p1-5 (<http://www.openssh.org>)
- OpenSSL 0.9.8g-15 (<http://www.openssl.org>)



## 1.2 RRDTool internals

### 1.2.1 The Round Robin Database

RRD is the acronym for Round Robin Database. RRD is a system to store and display time series data (i.e. network bandwidth, machine-room temperature, server load average etc). It stores the data in a very compact format that will not expand over time, and it is able to generate good looking graphs.

The Round Robin mechanism dictates that once a database has been written full, the oldest data will be overwritten when new data is stored. The period after which the stored data is overwritten, is determined upon creation of the RRD. After RRD creation, its file size remains the same.

Each graph in ACP corresponds with one RRD file.

### 1.2.2 Data consolidation

The data that is shown in the capacity and performance graphs may be consolidated, depending on the selected period.

If a graph spans a period of no longer than the previous 8 days, it is based on primary data point measurements. The primary data point interval is fixed and is set to 10 minutes.

If the graph period goes further down in history than the previous 8 days, the graph will be based on consolidated measurement values based on a number of 10-minute measurements (primary data points). A consolidation function is used to calculate the aggregated value. ACP stores the results of an average, a minimum and a maximum consolidation function. The average function is used by default for displaying graphs.

The following consolidation levels have been chosen by design of ACP:

<i>Graph period</i>	<i>Number of primary data points for consolidation</i>	<i>Consolidation period</i>
0 to 8 days	1	10 minutes
8 days to 1 month	3	30 minutes
1 to 6 months	12	2 hours
6 months to 5 years	144	1 day

For example, a graph spanning a period of the last 6 months is based on 2 hour consolidated samples. One sample is the average, minimum or maximum value of 12 primary data points. The consolidation function dropdown box in the title frame is used to select the consolidation function used to view graphs.

In availability graphs, no consolidation is applied. Availability graphs for any period are always based on primary data points (10 minutes interval), up to a maximum of 500 days.

### 1.2.3 Round Robin Archives

A Round Robin Database contains one or more Round Robin Archives (RRAs). An RRA contains the time-series data for each of the defined metrics (data sources) at one consolidation level and for one consolidation function. For each consolidation level, ACP stores the consolidation functions MAX, AVERAGE and MIN. An RRD file can contain more than one metric (data source) for a given object (for example memory used and memory free for a windows server), that will be displayed together in the same graph.



**Example:**

For a storing memory statistics of a Windows server, 3 data sources are defined in the RRD file:

- Physical memory size
- Page file size
- Used memory size.

An RRA is created for each combination of consolidation function and consolidation level. Each RRA contains data values for each of the data sources:

<b>RRA</b>	<b>Consolidation level (# PDPs)</b>	<b>Consolidation function</b>
1	1	MIN
2	3	MIN
3	12	MIN
4	144	MIN
5	1	MAX
6	3	MAX
7	12	MAX
8	144	MAX
9	1	AVERAGE
10	3	AVERAGE
11	12	AVERAGE
12	144	AVERAGE

When a graph is generated, RRDTool automatically selects the best fitting RRA for the selected graph period and consolidation function.

For a more detailed description of RRDTool workings, please refer to <http://www.rrdtool.org> and <http://www.vandenbogaerdt.nl/rrdtool>.





## 2 ACP Architecture

### 2.1 Local storage infrastructure

In a default ACP setup, the application runs on a dedicated server and the RRD files are stored on the local file system. In most cases, this setup is sufficient for monitoring up to around 1000 objects. Depending on the ACP server's specifications, the object types monitored and number of graphs per object, this number may be lower.

### 2.2 Shared storage infrastructure

In some cases it is necessary to deploy multiple ACP servers that write and read RRD files on a shared storage device. Examples of these cases are:

- When objects to be monitored reside in separated networks
- When the number of monitored objects results in exhaustion of the ACP server's CPU capacity.

In The Netherlands, both cases apply. In the NL setup, there are two physically separated networks: the Logica corporate network (formerly known as Gecis) and the customers shared network (Beheer). Each network contains two ACP servers to be able to extend the number of monitored objects. All four servers act as fetcher nodes and write their data to a shared storage device (Netapp filer). All four servers are also accessible through their own web interface.

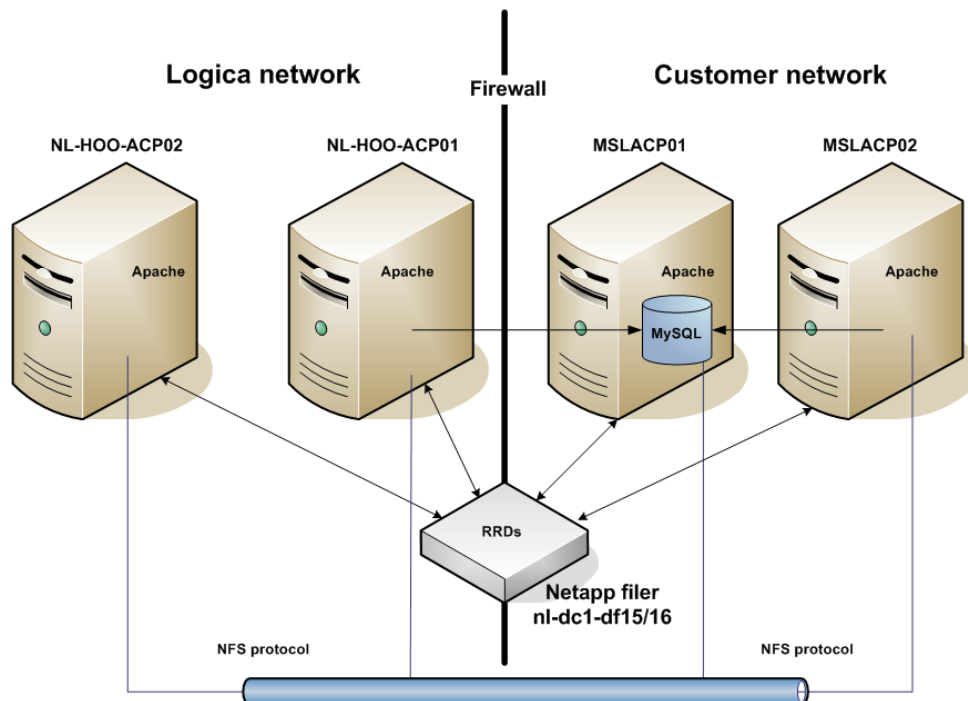


Figure 1 Shared storage ACP infrastructure in NL



## 2.3 MySQL database

### 2.3.1 Purpose

The ACP web application stores all information about monitored objects in a MySQL database, except for the metrics history which is stored in the RRD files. The database runs locally on the ACP server. Only the web interface connects to the database; the fetchers run independently from the web application.

### 2.3.2 Database location

Up to ACP v2.1, which uses a source installation of MySQL server, the MySQL database files are located in the directory `/usr/local/mysql/var`.

As from ACP v2.2, which uses a DEB package installation of MySQL server, the database file directory is specified in `/etc/mysql/my.cnf` and defaults to `/var/lib/mysql`.

The ACP database name is **acpmenu**.

### 2.3.3 ACP tables

The **acpmenu** database on each server contains the following tables:

Table	Description
Adminmenu	Contains the structure for the Administration menu
Chain_subtype	Contains definitions of object types present in each tier of an infra chain layer.
Chain_type	Contains definitions of layers present in a chain
Change_history	Contains information about object additions, modifications and deletions
Contracts	Contains the SLA contract types (Bronze, Silver, Gold)
Downtime_details	Contains downtime specifications (planned/unplanned; comment) for objects in ACP
Fetcher_nodes	Contains information about each ACP server and whether or not object additions are allowed on each server
Forbid_pwd	Contains forbidden passwords
Global_settings	Contains ACP portal global settings: <ul style="list-style-type: none"><li>• ACP base directory (default: <code>/home/acp/acp</code>)</li><li>• URL to bin directory (<code>http://&lt;servername&gt;/acp/bin</code>) - deprecated</li><li>• RRD files directory (default: <code>/home/acp/acp/rrd</code>)</li><li>• RRD files backup directory (default: <code>/home/acp/acp/rrdbackup</code>)</li><li>• RRDTOOL installation directory (default: <code>/usr</code>)</li><li>• Color settings for headings, table header background, table row background, table total background and table border</li><li>• Uptime threshold for SLA reports</li><li>• The default password for new ACP users</li><li>• The number of days of inactivity after an ACP account will be locked</li><li>• Whether or not to show a dropdown box for tree menu views</li></ul>
Hw_alerts	Holds active trend breach alerts (status 1) as well as inactive trend breach alerts (status 0) which aren't older than the prediction retention period of the RRD file.
Hw_alert_settings	Contains settings for trend breach alerting methods.
Hw_settings	Contains a parameter to enable/disable trend breach detection and default values for the Holt-Winters forecasting algorithm.



Table	Description
Icons	Contains filenames to icons for each object type. The images are located in /home/acp/acp/menu/menuicons
Infra_chain	Contains infra chain ID and name (per customer)
Infra_subchain	Contains object IDs for each object selected in an infra chain
Legend_text	Contains the text used in the introduction page of PDF reports.
Locations	Contains data center locations
Login_history	Speaks for itself
Logins	Contains ACP users
Menu	Contains the tree structure for the Modify menu
Object	Contains all object types that ACP supports
Object_config	Contains object parameters that are used to generate the config files
Othermenu	Contains the tree structure for the Other menu
Owner	Contains the owner (service group) of each object type
Phplayersmenu	Contains the tree structure for the Customers menu
Phplayersmenu_i18n	Deprecated.
Service_windows	Contains service window days and start & end times of non-24x7 contracts
Site_wide_outages	Contains time details and a description of ACP server outages
Site_wide_outage_cust	Contains customers affected per site wide outage
Statuses	Contains production statuses (dev, test, acceptance, production, etc)
Template	Contains user-defined templates (name, default period)
Templateitem	Contains object selections per template
Template_schedule	Contains templates scheduled for email
Template_schedule_options	Contains report options per scheduled template
User_privs	Contains information about the customers read only ACP users have access to

All tables used by dhtmlXtree (phplayersmenu, adminmenu, menu and othermenu) are self-referential by means of the id and parent\_id columns. The top level item for any tree menu corresponds to the row with id = 1. All subitems under this top level item must have parent\_id = 1 and a unique id.

Apart from tables, the database also contains a number of views (names starting with “v\_”). These are used for the different tree views that can be selected in the title frame.



### 2.3.4 The ACP directory structure

The ACP home directory contains the following files and subdirectories:

Directory	File	Contents
/var/www	index.php	ACP login page
	acp	Symbolic link to /home/acp/acp
/home/acp/acp	index.php	Main ACP page that contains the frameset for the title, menu and content frame
/home/acp/acp/auth		PHP Authorization scripts for logging on and logging off
/home/acp/acp/bin		<ul style="list-style-type: none"><li>ACP Fetcher scripts (Python)</li><li>ACP PHP scripts for graphs, availability calculations, management reports, portal administration</li></ul>
/home/acp/acp/conf		Configuration scripts, one for each object type
/home/acp/acp/doc		ACP User manual and Administrator manual in PDF format
/home/acp/acp/docroot		Scripts that are copied to the Apache docroot directory during installation
/home/acp/acp/help		PHP scripts for online help
/home/acp/acp/images		Images used in the title frame
/home/acp/acp/locks		Lock files for running fetchers
/home/acp/acp/log		Fetcher log files, separated in subdirectories for each object type
/home/acp/acp/menu		<ul style="list-style-type: none"><li>dhtmlXtree sources (AJAX tree menu)</li><li>ACP PHP scripts for modifying the tree menus</li><li>ACP PHP scripts generating the config files</li></ul>
/home/acp/acp/rrd		RRD files created by the fetcher scripts, separated in subdirectories for each object type and customer
/home/acp/acp/rrdbackup		Backed up RRD files of deleted objects
/home/acp/acp/spool		Temporary Oracle spool files
/home/acp/acp/sql		Oracle, SQL Server and MySQL SQL scripts used by the fetcher scripts
/home/acp/acp/tmp		Temporary files containing http arguments for each session and temporary files for PDF generation
/home/acp/acp/unix		Python scripts for Linux and Unix fetchers
/home/acp/acp/upg		ACP Upgrade scripts and packages
/home/acp/acp/wmi		NRPE Executables and VBS scripts to be distributed to Windows servers that are monitored through the WMI interface
/home/acp/acp/xml		Output XML files from Linux and Unix fetchers



## 2.4 Starting and stopping ACP

All necessary services are started automatically upon system boot and stopped automatically upon system shutdown. During normal operation there is no need to manually stop and start services.

To manually start all ACP services follow the actions below:

1. Start MySQL
  - a. login on the ACP server as user root
  - b. `/etc/init.d/mysql start`
2. Start Apache
  - a. login on the ACP server as user root
  - b. `/etc/init.d/apache2 start`

To manually stop all ACP services follow the actions below:

3. Stop MySQL
  - a. login on the ACP server as user root
  - b. `/etc/init.d/mysql stop`
4. Stop Apache
  - a. login on the ACP server as user root
  - b. `/etc/init.d/apache2 stop`

## 2.5 Backup and restore

### 2.5.1 Linux

The following directories should be backed up:

```
/home/acp/acp  
/home/acp/acp/rrd
```

By default, no backup procedure is implemented. This should be done following local standards and procedures.

### 2.5.2 MySQL

The MySQL database should be backed up. Creating a dump file should be sufficient:

```
mysqldump --all-databases -u root -p --result-file=/home/acp/acp/rrd/acp_db.dmp
```

The restore is performed by importing the dump file:

```
mysql -u root -p < acp_db.dmp
```

An alternative to the dump is bringing the MySQL server down and to backup the physical database files that reside in `/usr/local/mysql/var` (MySQL source install used up to ACP v2.1) or `/var/lib/mysql` (MySQL deb package install used as from ACP v2.2).

Make sure the dump file or the physical backup is included in the system backup.



## 3 Additional configuration

### 3.1 ACP configuration file

The ACP application configuration file is `/etc/acp.conf`. Actually `/etc/acp.conf` is a symbolic link to `/home/acp/acp/conf/acp.conf`.

The ACP configuration file is read by both the data collection scripts (fetchers) and the web interface. It contains the following sections:

- A global parameter section
- A client software environment variables section
- A parameter section for each object type.

A template of `acp.conf` is available in `/home/acp/acp/conf/acp.conf.template`.

#### 3.1.1 Global section

```
# Global section
#
# Directory where rrdtool is installed
RRD=/usr
# Basedirectory where the acp tool lives
ACPBASE=/home/acp/acp
# Email addresses of ACP administrators (comma separated)
ACP_ADMINS="etaacpsupport@logica.com"
SEND_EMAIL_THROUGH_RELAY=yes
# Proxy settings (leave commented when a proxy should not be used)
HTTP_PROXY=http://158.234.250.71:80/
# umask
umask 002
```

An example of the global section in `acp.conf`. This section contains the following parameters:

- **RRD**  
The installation directory of `rrdtool`, by default `/usr`. The `rrdtool` binaries reside in `/usr/bin`. The ACP fetcher scripts assume the `bin` subdirectory to be present under `/usr`.
- **ACPBASE**  
The installation directory of the ACP application, by default `/home/acp/acp`.
- **ACP\_ADMINS**  
A comma separated list of ACP administrators managing this ACP instance. This defaults to `etaacpsupport@logica.com`, the ACP global support team email address.
- **SEND\_EMAIL\_THROUGH\_RELAY** (yes/no)  
This parameter indicates whether or not ACP can send out emails through an email relay server present in the network. If set to yes, the `exim4` service on the ACP server must be configured to use a relay server (see paragraph 3.3) and the email relay itself must be configured to accept emails from the ACP server.
- **HTTP\_PROXY**  
When specified, ACP tries to use this proxy for all outgoing HTTP connections. This applies to both the fetchers and the web interface.
- **HTTPS\_PROXY**  
Similar to `HTTP_PROXY`, only for HTTPS connections.
- **umask**  
The umask used by the fetchers when creating files and directories.



### 3.1.2 Client software environment variables section

```
# Oracle environment variables
#
# Directory where oracle software is installed
ORACLE_HOME=/oracle/client/9.2.0.4
PATH=/usr/bin:/usr/local/bin:$ORACLE_HOME/bin:$PATH
# Directory where tnsnames.ora is located
TNS_ADMIN=/oracle/net/network/admin
export ORACLE_HOME PATH TNS_ADMIN

# Mysql environment variables
#
# Directory where mysql is installed
MY_HOME=/usr
```

This section contains environment variables for the Oracle client and MySQL client. These parameters should not be modified during normal operation.

### 3.1.3 Parameters per object type section

```
# Apache fetcher settings
#
# verbosity (0=no log, 1=least, 4=most)
APACHE_VERBOSITY=1
# hping enabled (0=no, 1=yes)
APACHE_HPING_ENABLED=1
# output to logfile (0=STDOUT, 1=logfile)
APACHE_LOG=1
```

An example of the parameter section for Apache. Each object type has parameters for fetcher log verbosity (default 1), HPing (Hyperping) enabled (default 1) and logging (default 1).

If you change any of these parameters, this will apply to all fetcher runs for all customers.

## 3.2 DNS

Hostname lookups through DNS are configured in `/etc/resolv.conf`.

```
search groupinfra.com logica.com
nameserver 10.20.191.95
nameserver 10.19.191.95
```

An example of `/etc/resolv.conf`. The search parameter is used to specify a list of DNS domains to search when doing hostname lookups. For each DNS server to use for lookups (max 3), specify the nameserver parameter on a separate line.

If DNS is not available, a list of IP addresses and hostnames should be maintained in `/etc/hosts`.



### 3.3 Email

To enable ACP to send out emails, the exim4 service should be configured to use an external email relay server. This is done in `/etc/exim4/update-exim4.conf.conf`.

```
# /etc/exim4/update-exim4.conf.conf
#
# Edit this file and /etc/mailname by hand and execute update-exim4.conf
# yourself or use 'dpkg-reconfigure exim4-config'
#
# Please note that this is _not_ a dpkg-conffile and that automatic changes
# to this file might happen. The code handling this will honor your local
# changes, so this is usually fine, but will break local schemes that mess
# around with multiple versions of the file.
#
# update-exim4.conf uses this file to determine variable values to replace
# the DEBCONFsomethingDEBCONF strings in the configuration template files.
#
# Most settings found in here do have corresponding questions in the
# Debconf configuration, but not all of them.
#
# This is a Debian specific file

dc_eximconfig_configtype='satellite'
dc_other_hostnames=''
dc_local_interfaces='127.0.0.1'
dc_readhost='logica.nl'
dc_relay_domains=''
dc_minimaldns='false'
dc_relay_nets=''
dc_smarthost='mailgateway'
CFILEMODE='644'
dc_use_split_config='false'
dc_hide_mailname='true'
dc_mailname_in_oh='true'
dc_localdelivery='mail_spool'
```

When configuring exim4, make sure to set the following parameters:

- `dc_eximconfig_configtype='satellite'`
- `dc_readhost`  
Enter a valid domain name (for example: `dc_readhost='logica.com'`)
- Enter a valid email gateway hostname or IP for `dc_smarthost` (for example:  
`dc_smarthost='mailgateway'`)

After changing `/etc/exim4/update-exim4.conf.conf` restart the exim4 service:  
`/etc/init.d/exim4 restart`

NOTE: the email relay server specified must be configured to accept emails from the ACP server in order for emails to be sent through.





### 3.4 Time zone

ACP's data is stored as a time series data and therefore it is essential that the time zone is set correctly on the ACP server.

To change the time zone of the ACP server:

```
dpkg-reconfigure tzdata
```

To check the time zone on the ACP server:

```
cat /etc/timezone
```

### 3.5 Network Time Protocol

ACP's data is stored as a time series data and therefore it is essential that the time is set correctly on the ACP server.

The preferred method of setting the time is to keep it synchronized with an NTP server. To be able to do that the ntp Debian package needs to be installed and `/etc/ntp.conf` needs to be configured.

In `/etc/ntp.conf`, find the NTP server section and enter each NTP server to be used on a separate line, for example:

```
server 172.15.255.253  
server 127.127.1.0
```

After changing `/etc/ntp.conf` restart the NTP daemon:

```
/etc/init.d/ntp restart
```



## 4 Fetchers

### 4.1 Fetchers and config files

A fetcher is a script that collects data for a number of objects of a specific type and writes the collected data to RRD files. The method of data collection differs depending on the object type. All fetchers scripts are Python scripts.

A fetcher accepts one mandatory argument, the fetcher name, and one optional argument, the log level:

```
/home/acp/acp/bin/acp_fetch_<objecttype>.py <fetchername> <loglevel>
```

- If no fetcher name is supplied, the script exits.
- Log level ranges from 1 to 4. If no log level is supplied, a value of 1 is assumed.

The fetcher name supplied needs to be present in the config file of the object type:

```
/home/acp/acp/conf/<objecttype>.conf
```

In the config file, all objects (of all customers) of that object type are listed. For each object in the config file, a fetcher name is defined. When a fetcher runs, it reads through the config file and only interrogates objects having the supplied fetcher name. By using this construction, you can group objects that will be polled together in one run of a fetcher by a specific fetcher name.

**When an object is added or modified in the ACP web interface, the config file is regenerated automatically. There is no need to manually edit config files.**

Usually, the customer name is used for the fetcher name. However, if the completion time of a fetcher run exceeds the measurement interval of 10 minutes, it is necessary to divide the group of objects into multiple fetchers. To add a new fetcher name, simply enter a non-existing name when adding or modifying an object. Below is an example of a portion of the windows config file.

```
# windows.conf
#
# Generated by ACP portal on 21 Oct 2005 14:18
#
# Hostname      SNMP      SNMP      Customer      Fetcher
#              version   community
#####
# Customer: abn-amro
web-abn-01      2c          public      abn-amro      abn-amro
citrix-abn-01   2c          public      abn-amro      abn-amro
# Customer: ah
web-ah-02       2c          public      ah             ah
# Customer: athlon
athlm010        2c          public      athlon         athlon
athlm001        2c          public      athlon         athlon
athlm007        2c          public      athlon         athlon
athlm006        2c          public      athlon         athlon
athlm008        2c          public      athlon         athlon2
athlm009        2c          public      athlon         athlon2
athlm011        2c          public      athlon         athlon2
```

In this example, the servers named athlm010, athlm001, athlm007 and athlm006 will be interrogated by the fetcher run having fetcher name “athlon” and the servers named athlm008, athlm009 and athlm011 will be interrogated by the fetcher run having fetcher name “athlon2”.



All fetchers follow a similar procedure for collecting and storing data:

1. Check if the parameter for the fetcher name is set. If it is not, abort.
2. Read environment variables from the global ACP config file: `/etc/acp.conf`
3. Read the config file for the object type from `/home/acp/acp/conf/<objecttype>.conf`
4. For each object having the fetcher name supplied as the first argument of the fetcher script, perform the following steps:
  - a. hping the object with its specific protocol/port number. If there is no response, continue to the next object. The resulting measurement value for this object will be **unknown**. Hping2 requires root privileges, so the script executes this with `sudo`. This step is not implemented for all object types.
  - b. Check if the object is up. If it is, store the **up** value (0) in the RRD file and continue with the following checks for this object. If it is not, store the **down** value (1) in the RRD file and continue with the next object.

If an RRD file does not exist, the fetcher creates one automatically.

## 4.2 Logging

All fetchers generate logging into two files:

- `/home/acp/acp/log/general.log`  
To this log file all error conditions are written. It also contains scheduled report generation and email actions of templates.
- `/home/acp/acp/log/<object type>/<fetcher name>.log`  
To this log file the data collection and storage process is recorded. This may also include error conditions. This logging can be done using four log levels, level 4 being the most verbose. The log level is defined in the fetcher script itself.

## 4.3 ACP RRD files

The RRD files the ACP fetchers create are located in `/home/acp/acp/rrd`.

Since ACP v2.1, the primary location in which fetchers store RRD files, is:

**`/home/acp/acp/rrd/<objecttype>/<customer>`**

The change in location was necessary to avoid disk performance degradation when large numbers of RRD files are present. The new location is now leading. When an RRD file can not be found in the new location, ACP tries to find it in the old location (`/home/acp/acp/rrd`).

In ACP there is a distinction between static and dynamic graphs. A static graph generates one image per graph type. For example, an Uptime graph always generates one image. A dynamic graph generates multiple images per graph type. For example, a Network Interface Card graph generates a graph for each NIC present on a device.

The naming convention of static RRD files is:

`<customer>_<objectname>_<objecttype>_<graphname>.rrd`

The naming convention of dynamic RRD files is:

`<customer>_<objectname>_<objecttype>_<graphname>_<n>.rrd`

`<n>` can be a number or a name, depending on the object type.



## 4.4 Scheduling

The fetchers are scheduled in the crontab of the acp user. The 10 minute measurement interval is defined here. Specifying another measurement interval than 10 minutes will lead to unpredictable results in the graphs, because all RRD files are created with the same 10 minute interval (heart beat).

**When an object is added or modified in the ACP web interface, the crontab is regenerated automatically.**

ACP automatically balances the number of fetchers scheduled for execution over time.

When an object is added or modified, the new crontab contents are first written to a file. Then the acp crontab is updated with this file: `/home/acp/acp/acp-cron.<ACP_hostname>`

Below is an example of a portion of the ACP crontab entries:

```
#MIN    HR      DOM      MON      DOW      COM
#
# fetcher section
#
0,10,20,30,40,50 * * * * * /home/acp/acp/bin/acp_fetch_oracle.py rodamco
1,11,21,31,41,51 * * * * * /home/acp/acp/bin/acp_fetch_oracle.py pathe
2,12,22,32,42,52 * * * * * /home/acp/acp/bin/acp_fetch_oracle.py vrom
3,13,23,33,43,53 * * * * * /home/acp/acp/bin/acp_fetch_oracle.py commit
4,14,24,34,44,54 * * * * * /home/acp/acp/bin/acp_fetch_oracle.py oms
6,16,26,36,46,56 * * * * * /home/acp/acp/bin/acp_fetch_oracle.py hdms1
7,17,27,37,47,57 * * * * * /home/acp/acp/bin/acp_fetch_oracle.py logica
8,18,28,38,48,58 * * * * * /home/acp/acp/bin/acp_fetch_oracle.py gemams
9,19,29,39,49,59 * * * * * /home/acp/acp/bin/acp_fetch_oracle.py ah
0,10,20,30,40,50 * * * * * /home/acp/acp/bin/acp_fetch_oracle.py ech
1,11,21,31,41,51 * * * * * /home/acp/acp/bin/acp_fetch_oracle.py shell
2,12,22,32,42,52 * * * * * /home/acp/acp/bin/acp_fetch_oracle.py kluwer
3,13,23,33,43,53 * * * * * /home/acp/acp/bin/acp_fetch_oracle.py ttp
```

## 4.5 ACP maintenance jobs

Several other jobs are scheduled in the ACP crontab for various purposes:

```
# removal of logs
0 12 * * 0 /home/acp/acp/bin/trunklogs.bash

# Cleanup temp files older than one day in /acp/tmp directory
0 1 * * * find /acp/tmp -name "acp_tmp_*" -mtime +1 -exec rm -f {} \;

# kill long running fetchers (running for more than 30 minutes)
0,10,20,30,40,50 * * * * * /home/acp/acp/bin/kill_long_running_fetchers.bash
```

These ACP maintenance jobs are stored in a config file `/home/acp/acp/conf/acp-cron-maint.<ACP_hostname>`. This file is included in the crontab each time it is regenerated by ACP.

To modify the ACP maintenance jobs, follow these steps:

- Logon to the ACP server as user acp
- `cd /home/acp/acp/conf`
- `edit /home/acp/acp/conf/acp-cron-maint.<ACP_hostname>`
- Regenerate the crontab from the ACP web interface



## 5 Object types

### 5.1 Supported object types

Refer to the ACP User Manual for a list of supported object types.

### 5.2 How to configure objects for ACP monitoring

Most object types require additional configuration to enable ACP monitoring. The tables below list the configuration steps for each object type.

#### 5.2.1 Linux / Unix

To be able to use SSH public key authentication to remote Linux/Unix servers, an SSH public and private key pair is created for the `acp` user during ACP installation.

To manually create an SSH public key:

- Login as `acp`
- `ssh-keygen -t rsa`  
→ Press <Enter> when prompted for a pass phrase

This creates a hidden subdirectory `/home/acp/.ssh`, containing a public key `id_rsa.pub` and a private key `id_rsa`.



AIX	
Prerequisites	
<ul style="list-style-type: none"> <li>AIX v 4.3 or higher.</li> <li>SSH installed and SSH daemon running</li> <li>Automatic logon using public keys should be allowed by the SSH daemon (sshd_config): PubkeyAuthentication yes</li> <li>Any firewalls in between the ACP server and the monitored node should allow SSH traffic originating from the ACP server (TCP/22)</li> </ul>	
Configuration on monitored node*	Configuration on ACP portal*
<p>Login on the AIX server, as root.</p> <pre># mkgroup -'A' id='6778' acp # mkuser id='6778' pgrp='acp' home='/home/acp' shell='/usr/bin/ksh' groups='system' acp # mkdir /usr/users/acp # chown acp:acp /usr/users/acp # su - acp \$ mkdir acp .ssh \$ chmod 700 .ssh \$ cd .ssh \$ touch authorized_keys \$ chmod 600 authorized_keys \$ exit</pre> <p>Enter a password for user acp:</p> <pre># passwd</pre> <p>Communicate the acp password to the team responsible for ACP deployment</p>	<p>Login on the ACP server, as user acp</p> <pre>\$ vi ~/.ssh/id_rsa.pub</pre> <p>→ copy the ACP server's SSH public key to the clipboard and exit vi without saving</p> <pre>\$ ssh acp@&lt;AIX-hostname&gt;</pre> <p>→ type "yes" when asked to continue connecting</p> <pre>\$ cd .ssh \$ vi authorized_keys</pre> <p>→ go to insert mode; paste the ACP server's SSH public key from the clipboard; save and exit the file</p> <pre>\$ exit</pre> <p><b>For AIX v4.3 systems:</b></p> <pre>\$ cd acp/unix/aix/v4.x \$ scp -p aixsysinfo acp@&lt;AIX-hostname&gt;:acp</pre> <p>→ there should be no password prompt</p> <p><b>For AIX v5 or higher systems:</b></p> <pre>\$ cd acp/unix/aix/v5.x \$ scp -p aixsysinfo cpu dsk fs info lpar mem net nfs proc swap acp@&lt;AIX- hostname&gt;:acp</pre> <p>→ there should be no password prompt</p> <p>Test the script on the ACP portal server:</p> <pre>\$ ssh acp@&lt;AIX-hostname&gt; acp/aixsysinfo</pre>
	Adding an object
	<p>Specify the following mandatory object properties:</p> <ul style="list-style-type: none"> <li>Hostname</li> <li>Username (Unix user name; default: acp)</li> <li>Remote directory (default: acp)</li> <li>Contract</li> </ul>

\* This is the public key authentication setup for standard SSH. For Reflection SSH2 setup, refer to paragraph 5.2.2.



Linux	
Prerequisites	
<ul style="list-style-type: none"><li>Linux kernel 2.4 or higher</li><li>Python 2.2 or higher installed</li><li>SSH installed and SSH daemon running</li><li>Automatic logon using public keys should be allowed by the SSH daemon (sshd_config): PubkeyAuthentication yes</li><li>Any firewalls in between the ACP server and the monitored node should allow SSH traffic originating from the ACP server (TCP/22)</li></ul>	
Configuration on monitored node*	Configuration on ACP portal*
<p>Login on the Linux server, as root.</p> <pre># groupadd -g 6778 acp # useradd -u 6778 -c "ACP user" -g acp acp # su - acp \$ mkdir acp .ssh \$ chmod 700 .ssh \$ cd .ssh \$ touch authorized_keys \$ chmod 600 authorized_keys \$ exit</pre> <p>Enter a password for user acp:</p> <pre># passwd acp</pre> <p>Communicate the acp password to the team responsible for ACP deployment</p>	<p>Login on the ACP server, as user acp</p> <pre>\$ vi ~/.ssh/id_rsa.pub</pre> <p>→ copy the ACP server's SSH public key to the clipboard and exit vi without saving</p> <pre>\$ ssh acp@&lt;Linux-hostname&gt;</pre> <p>→ type "yes" when asked to continue connecting</p> <pre>\$ cd .ssh \$ vi authorized_keys</pre> <p>→ go to insert mode; paste the ACP server's SSH public key from the clipboard; save and exit the file</p> <pre>\$ exit \$ cd acp/unix/linux \$ scp -p *.py acp@&lt;Linux-hostname&gt;:acp</pre> <p>→ there should be no password prompt</p> <p>Test the script on the ACP portal server:</p> <pre>\$ ssh acp@&lt;Linux-hostname&gt; acp/linuxsysinfo.py</pre>
	Adding an object
	<p>Specify the following mandatory object properties:</p> <ul style="list-style-type: none"><li>Hostname</li><li>Username (Linux user name; default: acp)</li><li>Remote directory (default: acp)</li><li>Contract</li></ul>

\* This is the public key authentication setup for standard SSH. For Reflection SSH2 setup, refer to paragraph 5.2.2.



Sun Solaris	
Prerequisites	
<ul style="list-style-type: none"><li>• Solaris 8 or higher</li><li>• SSH installed and SSH daemon running</li><li>• Automatic logon using public keys should be allowed by the SSH daemon (sshd_config): PubkeyAuthentication yes</li><li>• Any firewalls in between the ACP server and the monitored node should allow SSH traffic originating from the ACP server (TCP/22)</li></ul>	
Configuration on monitored node*	Configuration on ACP portal*
<p>Login on the Solaris server, as root.</p> <pre># groupadd -g 6778 acp # useradd -u 6778 -c "ACP user" -g acp -s /usr/bin/bash -d /export/home/acp acp # mkdir /export/home/acp # chown acp:acp /export/home/acp # su - acp \$ mkdir acp .ssh \$ chmod 700 .ssh \$ cd .ssh \$ touch authorized_keys \$ chmod 600 authorized_keys \$ exit</pre> <p>Enter a password for user acp:</p> <pre># passwd acp</pre> <p>Communicate the acp password to the team responsible for ACP deployment</p>	<p>Login on the ACP server, as user acp</p> <pre>\$ vi ~/.ssh/id_rsa.pub</pre> <p>→ copy the ACP server's SSH public key to the clipboard and exit vi without saving</p> <pre>\$ ssh acp@&lt;Solaris-hostname&gt;</pre> <p>→ type "yes" when asked to continue connecting</p> <pre>\$ cd .ssh \$ vi authorized_keys</pre> <p>→ go to insert mode; paste the ACP server's SSH public key from the clipboard; save and exit the file</p> <pre>\$ exit \$ cd acp/unix/solaris \$ scp -p *info acp@&lt;Solaris-hostname&gt;:acp</pre> <p>→ there should be no password prompt</p> <p>Test the script on the ACP portal server:</p> <pre>\$ ssh acp@&lt;Solaris-hostname&gt; acp/solarissysinfo</pre>
	Adding an object
	<p>Specify the following mandatory object properties:</p> <ul style="list-style-type: none"><li>• Hostname</li><li>• Username (Unix user name; default: acp)</li><li>• Remote directory (default: acp)</li><li>• Contract</li></ul>

\* This is the public key authentication setup for standard SSH. For Reflection SSH2 setup, refer to paragraph 5.2.2.





HP-UX	
Prerequisites	
<ul style="list-style-type: none"><li>• HP-UX v.11 or higher</li><li>• SSH installed and SSH daemon running</li><li>• Automatic logon using public keys should be allowed by the SSH daemon (sshd_config): PubkeyAuthentication yes</li><li>• Any firewalls in between the ACP server and the monitored node should allow SSH traffic originating from the ACP server (TCP/22)</li></ul>	
Configuration on monitored node*	Configuration on ACP portal*
<p>Login on the HP-UX server, as root.</p> <pre># groupadd -g 6778 acp # useradd -u 6778 -c "ACP user" -g acp -s /usr/bin/ksh acp # mkdir /export/home/acp # chown acp:acp /home/acp # su - acp \$ mkdir acp .ssh \$ chmod 700 .ssh \$ cd .ssh \$ touch authorized_keys \$ chmod 600 authorized_keys \$ exit</pre> <p>Enter a password for user acp:</p> <pre># passwd acp</pre> <p>Communicate the acp password to the team responsible for ACP deployment</p>	<p>Login on the ACP server, as user acp</p> <pre>\$ vi ~/.ssh/id_rsa.pub</pre> <p>→ copy the ACP server's SSH public key to the clipboard and exit vi without saving</p> <pre>\$ ssh acp@&lt;HPUX-hostname&gt;</pre> <p>→ type "yes" when asked to continue connecting</p> <pre>\$ cd .ssh \$ vi authorized_keys</pre> <p>→ go to insert mode; paste the ACP server's SSH public key from the clipboard; save and exit the file</p> <pre>\$ exit \$ cd acp/unix/hpux \$ scp -p hpuxsysinfo swap acp@&lt;HPUX-hostname&gt;:acp</pre> <p>→ there should be no password prompt</p> <p>Test the script on the ACP portal server:</p> <pre>\$ ssh acp@&lt;HPUX-hostname&gt; acp/hpuxsysinfo</pre>
	Adding an object
	<p>Specify the following mandatory object properties:</p> <ul style="list-style-type: none"><li>• Hostname</li><li>• Username (Unix user name; default: acp)</li><li>• Remote directory (default: acp)</li><li>• Contract</li></ul>

\* This is the public key authentication setup for standard SSH. For Reflection SSH2 setup, refer to paragraph 5.2.2.



HP Tru64 / OSF	
Prerequisites	
<ul style="list-style-type: none"><li>• Tru64 v5.0 or higher</li><li>• SSH installed and SSH daemon running</li><li>• Automatic logon using public keys should be allowed by the SSH daemon (sshd_config): PubkeyAuthentication yes</li><li>• Any firewalls in between the ACP server and the monitored node should allow SSH traffic originating from the ACP server (TCP/22)</li></ul>	
Configuration on monitored node*	Configuration on ACP portal*
<p>Login on the Tru64 server, as root.</p> <pre># groupadd -g 6778 acp # useradd -u 6778 -c "ACP user" -g   acp -G kmem -s /usr/bin/ksh acp # mkdir /usr/users/acp # chown acp:acp /usr/users/acp # su - acp \$ mkdir acp .ssh \$ chmod 700 .ssh \$ cd .ssh \$ touch authorized_keys \$ chmod 600 authorized_keys \$ exit</pre> <p>Enter a password for user acp:</p> <pre># passwd acp</pre> <p>Communicate the acp password to the team responsible for ACP deployment</p>	<p>Login on the ACP server, as user acp</p> <pre>\$ vi ~/.ssh/id_rsa.pub</pre> <p>→ copy the ACP server's SSH public key to the clipboard and exit vi without saving</p> <pre>\$ ssh acp@&lt;Tru64-hostname&gt;</pre> <p>→ type "yes" when asked to continue connecting</p> <pre>\$ cd .ssh \$ vi authorized_keys</pre> <p>→ go to insert mode; paste the ACP server's SSH public key from the clipboard; save and exit the file</p> <pre>\$ exit \$ cd acp/unix/osf \$ scp -p def io netstat osfsysinfo proc   acp@&lt;Tru64-hostname&gt;:acp</pre> <p>→ there should be no password prompt</p> <p>Test the script on the ACP portal server:</p> <pre>\$ ssh acp@&lt;Tru64-hostname&gt; acp/osfsysinfo</pre>
	Adding an object
	<p>Specify the following mandatory object properties:</p> <ul style="list-style-type: none"><li>• Hostname</li><li>• Username (Unix user name; default: acp)</li><li>• Remote directory (default: acp)</li><li>• Contract</li></ul>

\* This is the public key authentication setup for standard SSH. For Reflection SSH2 setup, refer to paragraph 5.2.2.



Unix generic (SNMP)	
<b>Prerequisites</b>	
<ul style="list-style-type: none"> <li>• SNMP daemon installed running</li> <li>• Any firewalls in between the ACP server and the monitored node should allow SNMP traffic originating from the ACP server (default port UDP/161)</li> </ul>	
Configuration on monitored node	Configuration on ACP portal
<ul style="list-style-type: none"> <li>• Login to the Unix server as root</li> <li>• If SNMP requests are not allowed from all hosts, grant the ACP server (IP) SNMP read access</li> <li>• If ACP cannot use existing SNMP community strings, specify a new SNMP community string</li> <li>• (Re)start the SNMP daemon, if needed</li> </ul>	No configuration needed
	<b>Adding an object</b>
	Specify the following mandatory object properties: <ul style="list-style-type: none"> <li>• Hostname</li> <li>• SNMP version</li> <li>• SNMP community string (default: public)</li> <li>• Contract</li> </ul>

### 5.2.2 Reflection SSH2 setup

In case Reflection SSH2 is used on the remote Linux/Unix server instead of the standard SSH suite, the configuration on the ACP portal section needs to follow this procedure:

Configuration on monitored node*	Configuration on ACP portal
Setup the group acp and user acp following the instructions for the Unix/Linux flavour on the monitored node. Then follow this procedure: <pre># su - acp \$ mkdir acp .ssh2 \$ chmod 700 .ssh2 \$ cd .ssh2 \$ touch name.pub \$ chmod 600 name.pub \$ touch authorizations \$ chmod 600 authorizations \$ exit</pre> Enter a password for user acp: <pre># passwd acp</pre> Communicate the acp password to the team responsible for ACP deployment	Login on the ACP server, as user acp <pre>\$ vi ~/.ssh/id_rsa.pub</pre> → copy the ACP server's SSH public key to the clipboard and exit vi without saving <pre>\$ ssh acp@&lt;unix-hostname&gt;</pre> → type "yes" when asked to continue connecting <pre>\$ cd .ssh2 \$ vi name.pub</pre> → go to insert mode; paste the ACP server's SSH public key from the clipboard; save and exit the file <pre>\$ ssh-keygen -O name.pub (capital letter O)</pre> → this creates name.pub.ssh2 <pre>\$ vi authorization</pre> → go to insert mode and enter: key name.pub.ssh2 → save and exit the file <pre>\$ exit</pre> From this point onwards, follow the instructions for the specific Unix flavour to scp the necessary files to the remote Linux/Unix server.



### 5.2.3 Microsoft

Microsoft Windows (SNMP)	
Prerequisites	
<ul style="list-style-type: none"><li>Windows NT Server 4.0, 2000 Server, 2003 Server or higher</li><li>Windows server Administrator password or access to an account with administrator privileges</li><li>SNMP installed and SNMP service running</li><li>Any firewalls in between the ACP server and the monitored node should allow SNMP traffic originating from the ACP server (default port UDP/161)</li></ul>	
Configuration on monitored node	Configuration on ACP portal
<ul style="list-style-type: none"><li>Login on the Windows server with administrator privileges</li><li>Open the SNMP service properties (Security tab)</li><li>Specify a read only SNMP community string</li><li>Check the bullet "Accept SNMP packets from these hosts"</li><li>Click Add to specify the ACP server's hostname or IP address</li><li>(Re)start the SNMP service, if needed</li></ul>	No configuration needed Use SNMP version 1 for NT servers or if SNMP version 2c calls return incomplete data.
	Adding an object
	Specify the following mandatory object properties: <ul style="list-style-type: none"><li>Hostname</li><li>SNMP version</li><li>SNMP community string (default: public)</li><li>Contract</li></ul>

Microsoft Windows (WMI)	
Prerequisites	
<ul style="list-style-type: none"><li>Windows 2000 Server, 2003 Server or higher</li><li>Windows server Administrator password or access to an account with administrator privileges</li><li>A ZIP file containing all sources from the subdir <code>/home/acp/acp/wmi</code> on the ACP server</li><li>A method of uploading the zip file to the remote Windows server</li><li>Any firewalls in between the ACP server and the monitored node should allow traffic originating from the ACP server to the Windows server (default port TCP/5666)</li></ul>	
Configuration on monitored node	Configuration on ACP portal
<ul style="list-style-type: none"><li>Login on the Windows server with administrator privileges</li><li>Create the directory <code>C:\nrpe_nt\bin</code></li><li>C:</li><li><code>cd \nrpe_nt\bin</code></li><li>Transfer the ZIP file to this directory and unzip it</li><li>Edit <code>nrpe.cfg</code>: Add the ACP server IP address to <code>allowed_hosts</code> (comma delimited list) If needed, change the <code>server_port</code> (default 5666). If you do, firewall rules should be changed accordingly.</li><li><code>NRPE_NT.exe -i</code> This installs the service "Nagios Remote Plugin Executor for NT/W2K"</li><li>Go to Services and start the service.</li><li>Set the service to start automatically upon boot</li></ul>	No configuration needed
	Adding an object
	Specify the following mandatory object properties: <ul style="list-style-type: none"><li>Hostname</li><li>Port (default: 5666)</li><li>Contract</li></ul>



Microsoft Exchange Server (WMI)	
Prerequisites	
<ul style="list-style-type: none"><li>• Windows 2000 Server, 2003 Server or higher</li><li>• MS Exchange Server 2003 or higher</li><li>• Windows server Administrator password or access to an account with administrator privileges</li><li>• A ZIP file containing all sources from the subdir <code>/home/acp/acp/wmi</code> on the ACP server</li><li>• A method of uploading the zip file to the remote Windows server</li><li>• Any firewalls in between the ACP server and the monitored node should allow traffic originating from the ACP server to the Windows server (default port TCP/5666)</li></ul>	
Configuration on monitored node	Configuration on ACP portal
<ul style="list-style-type: none"><li>• Login on the Windows server with administrator privileges</li><li>• If no MS Exchange performance counters are present in Windows Performance monitor, enable the AutoDiscovery/AutoPurge (ADAP) process, which transfers performance libraries to the WMI repository: <code>wmiadap.exe /f</code></li><li>• Go to Services, start the service named "WMI Performance Adapter" and set it to start automatically upon boot</li><li>• Create the directory <code>C:\nrpe_nt\bin</code></li><li>• <code>C:</code></li><li>• <code>cd \nrpe_nt\bin</code></li><li>• Transfer the ZIP file to this directory and unzip it</li><li>• Edit <code>nrpe.cfg</code>: Add the ACP server IP address to <code>allowed_hosts</code> (comma delimited list) If needed, change the <code>server_port</code> (default 5666). If you do, firewall rules should be changed accordingly.</li><li>• <code>NRPE_NT.exe -i</code> This installs the service "Nagios Remote Plugin Executor for NT/W2K"</li><li>• Go to Services and start the service.</li><li>• Set the service to start automatically upon boot</li></ul>	No configuration needed
	<b>Adding an object</b>  Specify the following mandatory object properties: <ul style="list-style-type: none"><li>• Hostname</li><li>• Port (default: 5666)</li><li>• Contract</li></ul>



## 5.2.4 Virtualization

VMWare ESX server (SNMP)	
<b>Prerequisites</b>	
<ul style="list-style-type: none"><li>• SNMP daemon installed running on the physical server</li><li>• VMWare ESX server 2.x or higher; VMWare ESXi 3.x or higher</li><li>• A valid read only SNMP community string is available</li><li>• Any firewalls in between the ACP server and the physical server should allow SNMP traffic originating from the ACP server (default port UDP/161)</li></ul>	
Configuration on monitored node	Configuration on ACP portal
<ul style="list-style-type: none"><li>• If SNMP requests are not allowed from all hosts, grant the ACP server (IP) SNMP read access</li><li>• If ACP cannot use existing SNMP community strings, specify a new SNMP community string</li><li>• (Re)start the SNMP daemon, if needed</li></ul>	No configuration needed
	<b>Adding an object</b>
	Specify the following mandatory object properties: <ul style="list-style-type: none"><li>• Hostname</li><li>• SNMP version</li><li>• SNMP community string (default: public)</li><li>• Contract</li></ul>

VMWare Virtual Machine	
<b>Prerequisites</b>	
<ul style="list-style-type: none"><li>• VMWare ESX server object added and responding to ACP</li><li>• VMWare ESX fetcher must have at least run once. After that ACP, automatically detects all virtual machines present under the ESX server.</li></ul>	
Configuration on monitored node	Configuration on ACP portal
No additional configuration needed.	No configuration needed
	<b>Adding an object</b>
	Specify the following mandatory object properties: <ul style="list-style-type: none"><li>• Select VMWare ESX server</li><li>• Select virtual machine</li></ul>



## 5.2.5 Databases

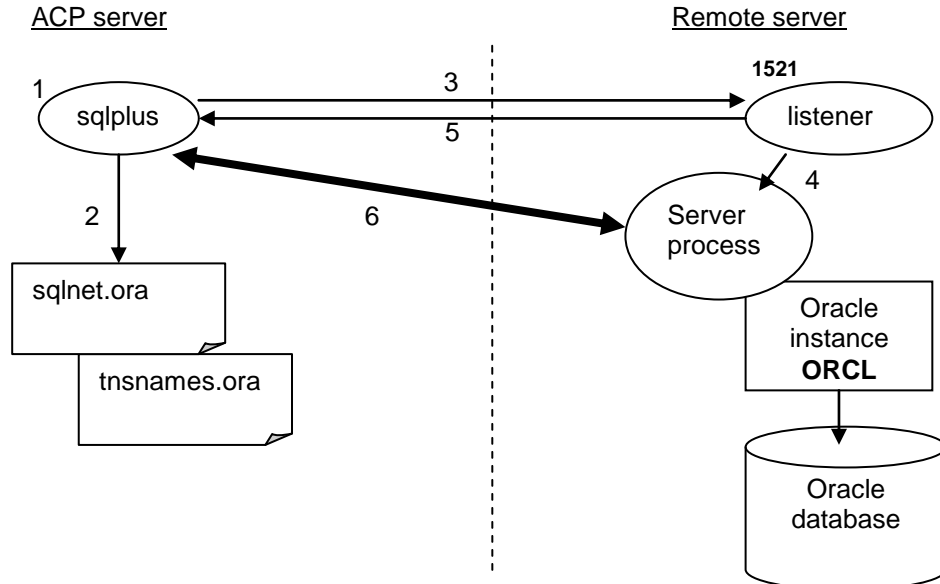
MySQL	
Prerequisites	
<ul style="list-style-type: none"><li>MySQL Server v3.23 or higher</li><li>Any firewalls in between the ACP server and the monitored node should allow MySQL traffic originating from the ACP server (default port TCP/3306)</li></ul>	
Configuration on monitored node	Configuration on ACP portal
<ul style="list-style-type: none"><li>Logon to the target MySQL database as root</li><li>Create a separate user <code>acp@&lt;ACP server's IP&gt;</code></li><li>Grant select and execute privileges to <code>acp@&lt;ACP server's IP&gt;</code></li><li>Communicate the acp password to the team responsible for ACP deployment</li></ul>	No configuration needed
	Adding an object
	Specify the following mandatory object properties: <ul style="list-style-type: none"><li>Hostname</li><li>Port (default: 3306)</li><li>Database username (default: acp)</li><li>Database password</li><li>Database password confirmation</li><li>Contract</li></ul>

PostgreSQL	
Prerequisites	
<ul style="list-style-type: none"><li>PostgreSQL Server v8.1 or higher</li><li>Any firewalls in between the ACP server and the monitored node should allow PostgreSQL traffic originating from the ACP server (default port TCP/5432)</li></ul>	
Configuration on monitored node	Configuration on ACP portal
<ul style="list-style-type: none"><li>Grant the OS user that owns the PostgreSQL software (for example postgres) access to all databases in the configuration file <code>pg_hba.conf</code>: <pre>#TYPE DATABASE USER      CIDR-ADDRESS  METHOD host all          postgres &lt;ACP server IP&gt; md5</pre></li><li>Restart the PostgreSQL database</li></ul>	No configuration needed
	Adding an object
	Specify the following mandatory object properties: <ul style="list-style-type: none"><li>Hostname</li><li>Database name</li><li>Port (default: 5432)</li><li>Database username (default: acp)</li><li>Database password</li><li>Database password confirmation</li><li>Contract</li></ul>



### Introduction to Oracle client-server communications

Oracle clients request a connection to a remote Oracle instance with the Oracle listener that runs on the remote node. The Oracle client used by ACP is the standard Oracle utility sqlplus.



**Figure 2 Creating an Oracle client-server connection**

In order to establish an Oracle client-server connection, the following steps occur:

1. sqlplus is called with a TNS-alias (also known as connect string) by the Oracle fetcher  
For example: sqlplus acp@mydb
2. sqlplus tries to resolve the TNS-alias (mydb) in into:
  - a server name or IP
  - a port number
  - an instance name

There are different naming resolution methods. By default, a local file tnsnames.ora is used. On the ACP server, tnsnames.ora is located in /oracle/net/network/admin. For each TNS-alias one entry must be present in tnsnames.ora. This is an example of such an entry:

```
mydb =
  (DESCRIPTION=
    (ADDRESS_LIST=
      (ADDRESS= (PROTOCOL=TCP) (HOST=oraclehost01) (PORT=1521) )
    )
    (CONNECT_DATA= (SID=ORCL) )
  )
```

3. sqlplus issues a connection request on the hostname and port number found. The database listener must be configured for this port number and for instances it may service.
4. If the listener knows the instance in the connection request, it spawns a server process for the instance, which starts listening on a randomly selected port number above 1024.
5. The listener communicates the server process port number to the sqlplus and redirects the communication to this port. Firewalls in between must support this (SQLNET ruleset).
6. The final database connection is established between sqlplus and the server process on the server process port number.





Oracle database	
Prerequisites	
<ul style="list-style-type: none"><li>• Oracle Server v8.1 or higher</li><li>• Any firewalls in between the ACP server and the monitored node should allow SQL*Net v2 traffic (with SQL*Net rule set enabled for port redirection) originating from the ACP server (default port TCP/1521)</li></ul>	
Configuration on monitored node	Configuration on ACP portal
<ul style="list-style-type: none"><li>• Logon to the target Oracle database with SYSDBA privileges</li><li>• Create a separate user for acp: <code>create user acp identified by &lt;password&gt;;</code></li><li>• Grant privileges to acp: <pre>grant create session to ACP; grant select on dba_tablespace to ACP; grant select on dba_data_files to ACP; grant select on dba_temp_files to ACP; grant select on dba_free_space to ACP; grant select on dba_role_privs to ACP; grant select on v_\$sysstat to ACP; grant select on v_\$sysstat to ACP; grant select on v_\$session to ACP; grant select on v_\$system_event to ACP; grant select on v_\$parameter to ACP; grant select on v_\$database to ACP; grant select on v_\$instance to ACP; grant select on v_\$license to ACP; grant select on v_\$log to ACP; grant select on v_\$tempfile to ACP; grant select on v_\$datafile to ACP; grant select on v_\$temp_space_header to ACP; grant select on v_\$sort_usage to ACP; grant select on v_\$filestat to ACP; create or replace view x_\$kcbwh as select * from x\$kcbwh; create or replace view x_\$kcbsw as select * from x\$kcbsw; grant select on x_\$kcbwh to ACP; grant select on x_\$kcbsw to ACP;</pre></li><li>• Communicate the acp password to the team responsible for ACP deployment</li></ul>	<ul style="list-style-type: none"><li>• Login as acp</li><li>• <code>cd /oracle/net/network/admin</code></li><li>• <code>vi tnsnames.ora</code> → add an entry for the TNS-alias; save and exit the file</li></ul>
	Adding an object
	<p>Specify the following mandatory object properties:</p> <ul style="list-style-type: none"><li>• Instance (TNS alias)</li><li>• Database username (default: acp)</li><li>• Database password</li><li>• Database password confirmation</li><li>• Contract</li></ul>



Microsoft SQL Server	
Prerequisites	
<ul style="list-style-type: none"><li>SQL Server v7 or higher</li><li>Any firewalls in between the ACP server and the monitored node should allow TDS traffic originating from the ACP server (default port TCP/1433)</li><li>Security authentication mode in SQL Server needs to be set to "SQL Server only" or "SQL Server and Windows". If it is set to "Windows only", the ACP user login will fail.</li></ul>	
Configuration on monitored node	Configuration on ACP portal
<ul style="list-style-type: none"><li>Login to the target SQL Server instances with sa privileges</li><li>Execute the following script to create an acp login with default password Str0ngP@ssw0rd. The user name and password may be adapted according to local standards.</li></ul> <pre>USE MASTER GO -- you can choose your own username and password! sp_addlogin 'acp', 'Str0ngP@ssw0rd' exec sp_altermessage 1205, 'WITH_LOG', 'true' -- go  IF (EXISTS (SELECT name FROM msdb.dbo.sysalerts WHERE name = N'Deadlock Detected'))     ---- Delete the alert with the same name.     EXECUTE msdb.dbo.sp_delete_alert @name = N'Deadlock Detected' BEGIN EXECUTE msdb.dbo.sp_add_alert @name = N'Deadlock Detected', @message_id = 1205, @severity = 0, @enabled = 1, @delay_between_responses = 60, @include_event_description_in = 5, @category_name = N'[Uncategorized]' END  GO  DECLARE @dbname VARCHAR(80) DECLARE @str VARCHAR(2000)  SELECT name INTO #Databases FROM master..sysdatabases WHERE     DATABASEPROPERTY(name, N'IsDetached') = 0         AND DATABASEPROPERTY(name, N'IsSuspect') = 0         AND DATABASEPROPERTY(name, N'IsOffline') = 0         AND DATABASEPROPERTY(name, N'IsInLoad') = 0         AND DATABASEPROPERTY(name, N'IsReadOnly') = 0 SELECT @dbname = MIN(name) FROM #Databases WHILE @dbname IS NOT NULL BEGIN     SELECT @str = 'USE [' + @dbname +         ']' + CHAR(10) + 'exec sp_adduser acp' -- replace acp with your login     PRINT @str     EXEC (@str)     PRINT ''     PRINT ''     IF lower(@dbname) = 'msdb'     BEGIN         SELECT @str = 'USE [' + @dbname +             ']' + CHAR(10) + 'exec sp_addrolemember ''db_datareader'', ''acp''' -- replace acp with your login         PRINT @str         EXEC (@str)     END     DELETE #Databases WHERE name = @dbname     SELECT @dbname = MIN(name) FROM #Databases END use master</pre>	No configuration needed
	<b>Adding an object</b> Specify the following mandatory object properties: <ul style="list-style-type: none"><li>Hostname</li><li>Select default or named instance</li><li>Instance name for named instance</li><li>Port (default: 1433)</li><li>SQL Server version</li><li>Database username (default: acp)</li><li>Database password</li><li>Database password confirmation</li><li>Contract</li></ul>



DROP TABLE #Databases

- Communicate the acp password to the team responsible for ACP deployment

## 5.2.6 Web servers

Apache	
Prerequisites	
<ul style="list-style-type: none"><li>• Apache 1.3 or higher</li><li>• Any firewalls in between the ACP server and the monitored node should allow HTTP(S) traffic originating from the ACP server (default ports TCP/80 and TCP/443)</li></ul>	
Configuration on monitored node	Configuration on ACP portal
<ul style="list-style-type: none"><li>• Edit httpd.conf and set the following directives:<ul style="list-style-type: none"><li>◦ ExtendedStatus on</li><li>◦ &lt;Location /server-status&gt;<ul style="list-style-type: none"><li>SetHandler server-status</li><li>Order deny, allow</li><li>Deny from all</li><li>Allow from &lt;ACP server's IP&gt;</li></ul></li></ul>&lt;/Location&gt;</li><li>• Restart Apache</li></ul>	No configuration needed
	<b>Adding an object</b> Specify the following mandatory object properties: <ul style="list-style-type: none"><li>• Sitename</li><li>• Apache version</li><li>• Status URL (default: http://&lt;sitename&gt;/server-status)</li><li>• Use HTTP(S) proxy Proxy needs to be set in /etc/acp.conf</li><li>• Contract</li></ul>

Microsoft Internet Information Server (IIS)	
Prerequisites	
<ul style="list-style-type: none"><li>• Windows NT Server 4.0, 2000 Server, 2003 Server or higher</li><li>• SNMP installed and SNMP service running</li><li>• Any firewalls in between the ACP server and the monitored node should allow HTTP(S) traffic originating from the ACP server (default ports TCP/80 and TCP/443) and SNMP traffic originating from the ACP server (default port UDP/161)</li></ul>	
Configuration on monitored node	Configuration on ACP portal
<ul style="list-style-type: none"><li>• Login on the Windows server with administrator privileges</li><li>• Open the SNMP service properties (Security tab)</li><li>• Specify a read only SNMP community string</li><li>• Check the bullet "Accept SNMP packets from these hosts"</li><li>• Click Add to specify the ACP server's hostname or IP address</li><li>• (Re)start the SNMP service, if needed</li></ul>	No configuration needed
	<b>Adding an object</b> Specify the following mandatory object properties: <ul style="list-style-type: none"><li>• Hostname</li><li>• Status URL This can be any valid URL served by the web server</li><li>• Use HTTP(S) proxy Proxy needs to be set in /etc/acp.conf</li><li>• SNMP version</li><li>• SNMP community string (default:public)</li><li>• Contract</li></ul>



## 5.2.7 Application servers

Oracle iAS (identical to Apache)	
Prerequisites	
<ul style="list-style-type: none"><li>• Apache 1.3 or higher</li><li>• Any firewalls in between the ACP server and the monitored node should allow HTTP(S) traffic originating from the ACP server (default ports TCP/80 and TCP/443)</li></ul>	
Configuration on monitored node	Configuration on ACP portal
<ul style="list-style-type: none"><li>• Edit httpd.conf and set the following directives:<ul style="list-style-type: none"><li>◦ ExtendedStatus on</li><li>◦ &lt;Location /server-status&gt;<ul style="list-style-type: none"><li>SetHandler server-status</li><li>Order deny, allow</li><li>Deny from all</li><li>Allow from &lt;ACP server's IP&gt;</li></ul></li></ul></li><li>• Restart Apache</li></ul>	No configuration needed
	Adding an object
	<p>Specify the following mandatory object properties:</p> <ul style="list-style-type: none"><li>• Sitename</li><li>• Apache version</li><li>• Status URL (default: http://&lt;sitename&gt;/server-status)</li><li>• Use HTTP(S) proxy Proxy needs to be set in /etc/acp.conf</li><li>• Contract</li></ul>

Oracle OC4J	
Prerequisites	
<ul style="list-style-type: none"><li>• OC4J v9.0.4 or higher</li><li>• Any firewalls in between the ACP server and the monitored node should allow HTTP(S) traffic originating from the ACP server (default ports TCP/8888 and TCP/8889)</li></ul>	
Configuration on monitored node	Configuration on ACP portal
<ul style="list-style-type: none"><li>• Edit mod_oc4j.conf and set the following directives:<ul style="list-style-type: none"><li>◦ &lt;Location /dmsoc4j&gt;<ul style="list-style-type: none"><li>SetHandler oc4j-handler</li><li>Order deny, allow</li><li>Deny from all</li><li>Allow from &lt;ACP server's IP&gt;</li></ul></li></ul></li><li>• Restart Apache and OC4J</li></ul>	No configuration needed
	Adding an object
	<p>Specify the following mandatory object properties:</p> <ul style="list-style-type: none"><li>• Sitename</li><li>• Status URL (default: http://&lt;sitename&gt;/dmsoc4j/Spy)</li><li>• Use HTTP(S) proxy Proxy needs to be set in /etc/acp.conf</li><li>• Flag (default: JV)</li><li>• Contract</li></ul>



Tomcat	
Prerequisites	
<ul style="list-style-type: none"><li>• Tomcat v3.3 or higher</li><li>• Any firewalls in between the ACP server and the monitored node should allow HTTP(S) traffic originating from the ACP server (default ports TCP/80 and TCP/443)</li></ul>	
Configuration on monitored node	Configuration on ACP portal
<ul style="list-style-type: none"><li>• In the Tomcat admin page, specify that the admin pages can also be queried from the ACP server/IP.</li><li>• Restart Tomcat</li></ul>	No configuration needed
	Adding an object
	<p>Specify the following mandatory object properties:</p> <ul style="list-style-type: none"><li>• Sitename</li><li>• Status URL Specify as: http(s)://admin:&lt;pw&gt;@&lt;host&gt;:&lt;port&gt;/manager/status</li><li>• Use HTTP(S) proxy Proxy needs to be set in /etc/acp.conf</li><li>• Contract</li></ul>

JBoss	
Prerequisites	
<ul style="list-style-type: none"><li>• JBoss v3.2.6 or higher</li><li>• Any firewalls in between the ACP server and the monitored node should allow HTTP(S) traffic originating from the ACP server (default ports TCP/80 and TCP/443)</li></ul>	
Configuration on monitored node	Configuration on ACP portal
<ul style="list-style-type: none"><li>• In the JBoss admin page, specify that the admin pages can also be queried from the ACP server/IP.</li><li>• Restart Tomcat</li></ul>	No configuration needed
	Adding an object
	<p>Specify the following mandatory object properties:</p> <ul style="list-style-type: none"><li>• Sitename</li><li>• Status URL Specify as: http(s)://admin:&lt;pw&gt;@&lt;host&gt;:&lt;port&gt;/web-console/status</li><li>• Use HTTP(S) proxy Proxy needs to be set in /etc/acp.conf</li><li>• Contract</li></ul>



## 5.2.8 Network devices

The following applies for all network devices (routers, switches, firewalls, proxy servers, load balancers, VPNs etc):

Network devices (SNMP)	
Prerequisites	
<ul style="list-style-type: none"><li>• SNMP daemon installed running</li><li>• A valid read only SNMP community string is available</li><li>• Any firewalls in between the ACP server and the monitored node should allow SNMP traffic originating from the ACP server (default port UDP/161)</li></ul>	
Configuration on monitored node	Configuration on ACP portal
<ul style="list-style-type: none"><li>• If SNMP requests are not allowed from all hosts, grant the ACP server (IP) SNMP read access</li><li>• If ACP cannot use existing SNMP community strings, specify a new SNMP community string</li><li>• (Re)start the SNMP daemon, if needed</li></ul>	No configuration needed
	Adding an object
	Specify the following mandatory object properties: <ul style="list-style-type: none"><li>• Hostname</li><li>• SNMP version</li><li>• SNMP community string (default: public)</li><li>• Contract</li></ul>

## 5.2.9 Storage devices

Netapp filer	
Prerequisites	
<ul style="list-style-type: none"><li>• SNMP daemon installed running</li><li>• A valid read only SNMP community string is available</li><li>• Any firewalls in between the ACP server and the monitored node should allow SNMP traffic originating from the ACP server (default port UDP/161)</li></ul>	
Configuration on monitored node	Configuration on ACP portal
<ul style="list-style-type: none"><li>• If SNMP requests are not allowed from all hosts, grant the ACP server (IP) SNMP read access</li><li>• If ACP cannot use existing SNMP community strings, specify a new SNMP community string</li><li>• (Re)start the SNMP daemon, if needed</li></ul>	No configuration needed.
	Adding an object
	Specify the following mandatory object properties: <ul style="list-style-type: none"><li>• Hostname</li><li>• SNMP version Most Netapp filers use SNMP version 1</li><li>• SNMP community string (default: public)</li><li>• Contract</li></ul>



## 5.2.10 Directory servers

Fedora Directory Server (FDS)	
Prerequisites	
<ul style="list-style-type: none"><li>FDS v1.0.2 or v1.0.3</li><li>The password for the common name "Directory Manager" is known</li><li>Any firewalls in between the ACP server and the monitored node should allow LDAP traffic originating from the ACP server (default port TCP/389)</li></ul>	
Configuration on monitored node	Configuration on ACP portal
<ul style="list-style-type: none"><li>No configuration needed</li></ul>	No configuration needed
	Adding an object
	Specify the following mandatory object properties: <ul style="list-style-type: none"><li>Hostname</li><li>Port (default: 389)</li><li>Password</li><li>Password confirmation</li><li>Bind DN Specify as: <code>cn=Directory~Manager</code></li><li>Contract</li></ul>

## 5.2.11 General purpose

Ping (ICMP)	
Prerequisites	
<ul style="list-style-type: none"><li>Any firewalls in between the ACP server and the monitored node should allow ICMP traffic originating from the ACP server</li></ul>	
Configuration on monitored node	Configuration on ACP portal
<ul style="list-style-type: none"><li>No configuration needed</li></ul>	No configuration needed
	Adding an object
	Specify the following mandatory object properties: <ul style="list-style-type: none"><li>Hostname</li><li>Contract</li></ul>

TCP Ping	
Prerequisites	
<ul style="list-style-type: none"><li>A service or daemon running on the monitored node that listens on any TCP port number</li><li>Any firewalls in between the ACP server and the monitored node should allow TCP traffic on the specified port originating from the ACP server</li></ul>	
Configuration on monitored node	Configuration on ACP portal
<ul style="list-style-type: none"><li>No configuration needed</li></ul>	No configuration needed
	Adding an object
	Specify the following mandatory object properties: <ul style="list-style-type: none"><li>Hostname</li><li>Port</li><li>Contract</li></ul>



URL	
Prerequisites	
<ul style="list-style-type: none"><li>Any firewalls in between the ACP server and the monitored node should allow HTTP(S) traffic originating from the ACP server (default ports TCP/80 and TCP/443)</li></ul>	
Configuration on monitored node	Configuration on ACP portal
<ul style="list-style-type: none"><li>No configuration needed</li></ul>	No configuration needed
	Adding an object
	Specify the following mandatory object properties: <ul style="list-style-type: none"><li>Hostname</li><li>Status URL</li><li>Use HTTP(S) proxy Proxy needs to be set in /etc/acp.conf</li><li>Contract</li></ul>

Remote fetcher	
Prerequisites	
<ul style="list-style-type: none"><li>SSH installed and SSH daemon running</li><li>Automatic logon using public keys should be allowed by the SSH daemon (sshd_config): <code>PubkeyAuthentication yes</code></li><li>rsync 2.6.0 or higher installed on monitored node</li><li>Any firewalls in between the ACP server and the monitored node should allow SSH traffic originating from the ACP server (TCP/22)</li></ul>	
Configuration on monitored node	Configuration on ACP portal
<ul style="list-style-type: none"><li>Login on the Unix server, as root.</li><li>Add an acp user on the remote OS as described in paragraph 5.2.1.</li><li>Login as user acp and untar the archive containing the ACP sources.</li><li>Communicate the acp password to the team responsible for ACP deployment</li></ul>	Login on the ACP server, as user acp <pre>\$ vi ~/.ssh/id_rsa.pub</pre> → copy the ACP server's SSH public key to the clipboard and exit vi without saving <pre>\$ ssh acp@&lt;Linux-hostname&gt;</pre> → type "yes" when asked to continue connecting <pre>\$ cd .ssh</pre> <pre>\$ vi authorized_keys</pre> → go to insert mode; paste the ACP server's SSH public key from the clipboard; save and exit the file <pre>\$ exit</pre>
	Adding an object
	Add the object as its native type, not as remote.





Sendmail	
Prerequisites	
<ul style="list-style-type: none"><li>• SSH installed and SSH daemon running</li><li>• Automatic logon using public keys should be allowed by the SSH daemon (sshd_config): <code>PubkeyAuthentication yes</code></li><li>• Any firewalls in between the ACP server and the monitored node should allow SSH traffic originating from the ACP server (TCP/22)</li></ul>	
Configuration on monitored node	Configuration on ACP portal
<ul style="list-style-type: none"><li>• Login on the Unix server, as root.</li><li>• Add an acp user on the remote OS as described in paragraph 5.2.1.</li><li>• Communicate the acp password to the team responsible for ACP deployment</li></ul>	<p>Login on the ACP server, as user acp</p> <pre>\$ vi ~/.ssh/id_rsa.pub</pre> <p>→ copy the ACP server's SSH public key to the clipboard and exit vi without saving</p> <pre>\$ ssh acp@&lt;Linux-hostname&gt;</pre> <p>→ type "yes" when asked to continue connecting</p> <pre>\$ cd .ssh</pre> <pre>\$ vi authorized_keys</pre> <p>→ go to insert mode; paste the ACP server's SSH public key from the clipboard; save and exit the file</p> <pre>\$ exit</pre>
	<b>Adding an object</b> <p>Specify the following mandatory object properties:</p> <ul style="list-style-type: none"><li>• Hostname</li><li>• Username</li><li>• Remote command (default: /usr/bin/mailstats)</li><li>• Options (default: -p)</li><li>• Contract</li></ul>



## 6 Add & modify objects

The Modify menu in the web interface contains options to add, modify and delete objects from the web interface. This menu is only available for super users and administrators.

### 6.1 Adding a customer

When a customer is added, a short name and a full name have to be supplied for the customer.

- The full name is used as the label in the Customers menu.
- The short name is used as the customer identifier in URLs and RRD file names. It may not contain spaces.

### 6.2 Adding an object

In the Add Object screen, the user first has to select the customer under which the object will be added, the object type and the service level. Depending on the object type, different parameters have to be entered. These are specified in paragraph 5.2 for each individual object type.


All objects also need a fetcher name. Refer to paragraph 4.1 for a detailed description about fetchers and config files. The fetcher name that is entered will be scheduled in the ACP crontab. If the fetcher name already exists (listed under “Current fetchers” in the Add Object screen), the ACP crontab will not be altered and the added object will be fetched in the existing fetcher run by that name.

After adding an object:

- The config file for the object type is regenerated. This requires no further user action.
- The ACP crontab is regenerated to automatically include the fetcher name for this object type.
- The subtree containing the object type in the menu frame is refreshed automatically to show the new object in the menu. If the object can be reached and read with the specified protocol, graphs will appear within one measurement interval (10 minutes) after adding the object. Graph values will appear after two measurement intervals (20 minutes). If no graphs appear, or if the Uptime graph shows the object is down when in reality it is not, please contact the ACP administrator to investigate connectivity issues.



Figure 3 The add object screen for an Oracle database

An object can be added without enabling data collection. Data collection is enabled by default. To disable data collection, uncheck “Monitor this object”. Unmonitored objects will be displayed in the menu with a stop icon: 

The credentials supplied under “Object type specific settings” can be tested for connectivity by clicking “Test connection”. A popup screen will appear that shows the results of the connection test. This functionality does not work for SQL Server instances, for which the connection can only be tested after the object has been added.



## 6.3 Modifying or deleting an object

In the Modify Object screen, the user first has to select the customer and the object type under which the object resides. Depending on the object type, different parameters have to be filled in. Refer to paragraph 6.2 for the details.

In this screen the object can be deleted or modified. All parameters can be altered. The customer drop down box in the parameters section can be used to move an object to another customer.

Only ACP administrators are allowed to modify the Menu URL.

**Figure 4** The modify object screen of an Oracle database

After an object has been modified or deleted:

- The config file for the object type is regenerated. This requires no further user action.
- The ACP crontab is regenerated to automatically include the fetcher name for this object type.
- The subtree containing the object in the menu frame is refreshed automatically to reflect a change of object name, change of customer, or deletion of the object



## 7 Troubleshooting connectivity

### 7.1 Manual fetcher runs

Whenever a scheduled fetcher fails to collect data for one or more objects, you can try running the fetcher script manually from the command line with a high log level. Check the crontab to see the fetchers currently scheduled.

The log level can be 1 (default), 2, 3, or 4.

Example:

```
/home/acp/acp/bin/acp_fetch_windows.py osim 4
```

In this example, osim is the customer name and 4 is the log level. If no log level is given, a level of 1 is assumed. The customer name must be present in the config file of the fetcher for this object type, in this example in `/home/acp/acp/conf/windows.conf`.

```
acp@nl-hoo-acp01:~$ /acp/bin/acp_fetch_windows.py icube3 4
>> 17-Mar-2011 10:11:11 31610 Start of acp_fetch_windows.py, fetcher: icube3
>> 17-Mar-2011 10:11:11 31610 Logverbosity: 4
>> 17-Mar-2011 10:11:11 31610 RrdTimeStep: 600
>> 17-Mar-2011 10:11:11 31610 Pid: 31610
>> 17-Mar-2011 10:11:11 31610 Workingdirectory: /acp/bin
>> 17-Mar-2011 10:11:11 31610 Basedirectory: /home/acp/acp
>> 17-Mar-2011 10:11:11 31610 Logdirectory and file: /home/acp/acp/log/windows,
/home/acp/acp/log/windows/icube3.log
>> 17-Mar-2011 10:11:11 31610 RrdBasedirectory: /home/acp/acp/rrd/windows
>> 17-Mar-2011 10:11:11 31610 RrdBindirectory: /usr/bin
>> 17-Mar-2011 10:11:11 31610 Configdirectory and file: /home/acp/acp/conf,
/home/acp/acp/conf/windows.conf
>> 17-Mar-2011 10:11:11 31610 Lockdirectory and file: /home/acp/acp/locks,
/home/acp/acp/locks/windows.icube3.31610
>> 17-Mar-2011 10:11:11 31610 Statsdirectory, file and statslockfile: /home/acp/acp/stats,
/home/acp/acp/stats/nl-hoo-acp01-windows.stats, /home/acp/acp/stats/nl-hoo-acp01-
windows.stats.lock
>> 17-Mar-2011 10:11:11 31610 Objecttype: windows or WINDOWS
>> 17-Mar-2011 10:11:11 31610 Check for windows.icube3 lockfiles
>> 17-Mar-2011 10:11:11 31610 Create lockfile /home/acp/acp/locks/windows.icube3.31610
>> 17-Mar-2011 10:11:11 31610 Read configfile /home/acp/acp/conf/windows.conf
>> 17-Mar-2011 10:11:11 31610 Found object with fetcher icube3
>> 17-Mar-2011 10:11:11 31610 Fork proces to fetch data from nl-amv-gs03
>> 17-Mar-2011 10:11:11 31610 Child 31611 started
>> 17-Mar-2011 10:11:11 31610 Fetching windows information from nl-amv-gs03, SNMP version 2c,
customer: icube , fetcher: icube3
>> 17-Mar-2011 10:11:11 31610 Closed configfile /home/acp/acp/conf/windows.conf
>> 17-Mar-2011 10:11:11 31610 Wait for childs to exit
>> 17-Mar-2011 10:11:11 31610 Rrddirectory: /home/acp/acp/rrd/windows/icube
>> 17-Mar-2011 10:11:11 31611 Name or service nl-amv-gs03 can be resolved to 10.16.127.74
>> 17-Mar-2011 10:11:11 31611 Fetch memory MIB data, .1.3.6.1.2.1.25.3.3.1, from nl-amv-gs03
...
```

An example of a log level 4 manual fetcher run for Windows (output is clipped). The output allows you to track at which point the script fails. Any errors and warnings will be display as ERROR or WARNING.



## 7.2 Generic tests

In case an object does not respond, login to the Linux command prompt as user acp and perform the following checks:

`ping <hostname>`

- If this returns “unknown host” this indicates the host name is not known in DNS or the local hosts file on the ACP server
- If it succeeds, this indicates there is a route to the server
- If it fails, then a firewall could be blocking ICMP traffic.

Note that a successful or unsuccessful ping doesn’t tell you anything about whether TCP or UDP based protocols (SNMP, SSH, HTTP etc) will be (un)successful too, as ping uses ICMP.

`telnet <hostname> <port>`

- Telnet only works on TCP based protocols (SSH, HTTP(S), Oracle, SQL Server, MySQL, WMI)
- If you get a telnet prompt, then the firewall allows the connection and a process is listening at the specified port on the remote host.
- If you get “connection refused”, then the firewall allows the connection, but no process is listening at the specified port on the remote host.
- If you get a timeout, then the firewall blocks the connection.

## 7.3 SNMP

To test SNMP connectivity, login to the Linux command prompt as user acp and perform the following check:

`snmpstatus -v <version> -c <communitystring> <hostname>`

- The SNMP version can be either 1 or 2c
- The SNMP community string must be specified on the remote host

Since SNMP is UDP based, following an snmp request you either get an answer or you get a timeout. In case of a timeout there is no way of determining the cause of the timeout from the ACP server.

In case of a timeout, contact the technical team(s) to check:

- If a firewall allows SNMP traffic from the ACP server to the monitored host.
- If the SNMP version used to query the remote host is correct
- If the SNMP community string used to query the remote host is set on the SNMP daemon or service on the remote host (read only is sufficient).
- If the SNMP daemon or service on the remote host is configured to allow SNMP requests from the ACP server.



## 7.4 SSH

ACP uses SSH public key authentication to connect to remote (Unix/Linux) hosts. To test SSH connectivity to a remote host, login to the Linux command prompt as user acp and perform the following check:

```
ssh acp@<hostname>
```

- If it times out, then a firewall blocks SSH traffic from the ACP server to the remote host.
- If you get a password prompt, then SSH public key authentication is incorrectly configured. Recheck the configuration steps as listed in paragraph 5.2.1.
- If you are logged on without a password, check if the subdirectory “acp” exists on the remote host and if that contains the data collection scripts for the Unix flavor. If not, scp them from the ACP server (/home/acp/acp/unix/<objecttype>) to the remote host (~/.acp).

## 7.5 HTTP(S)

To test HTTP(S) connectivity to a remote host, login to the Linux command prompt as user acp and perform the following checks:

```
export http_proxy=http://[<user>/<password>@]<proxy-hostname>:<proxy-port>/
```

- Omit this command if no proxy is to be used
- If the proxy is a Microsoft ISA proxy, then configure the ntlmaps service (which is described in the ACP release notes) and set:  

```
export http_proxy=http://localhost:5865/
```

```
wget <url>
```

- If this times out then a firewall blocks HTTP(S) traffic from the ACP server to the remote host.
- If successful the URL page is downloaded.

## 7.6 Oracle

To test connectivity to a remote Oracle database, login to the Linux command prompt as user acp and perform the following checks:

```
. /etc/acp.conf
```

- This sets the Oracle client environment variables

```
tnsping <tns-alias>
```

- This reads /oracle/net/net/admin/tnsnames.ora and tries to contact the listener defined for this alias.
- If this returns “TNS-03505: Failed to resolve name” then the tns alias cannot be found in tnsnames.ora.
- If it times out, a firewall blocks traffic to the remote host and port number defined for this tns alias in tnsnames.ora.
- If it returns OK, then it also displays the TNS descriptor, showing the hostname and port number the database listener is listening on.



```
sqlplus <user>@<tns-alias>
```

- The user name and password for this tns alias are in `/home/acp/acp/conf/oracle.conf`.
- If it returns “ORA-12154: TNS:could not resolve service name” then the tns alias cannot be found in `tnsnames.ora`.
- If it times out, a firewall blocks traffic to the remote host and port number defined for this tns alias in `tnsnames.ora`.
- If the login is successful, enter the following query:  
`select * from v$database;`  
This should return a table row. If an error is returned then the user has insufficient privileges. Recheck the configuration steps in paragraph 5.2.5.

## 7.7 MS SQL Server

To test connectivity to a remote SQL Server instance, login to the Linux command prompt as user `acp` and perform the following checks:

```
bsqldb -U <username> -P <password> -S <servername> \  
-i /home/acp/acp/sql/uptime.sql -o out.txt -e err.txt
```

- The user name and password are in `/home/acp/acp/conf/sqlserver.conf`.
- The servername is in `/home/acp/acp/conf/freetds.conf`.
- Check `out.txt` and `err.txt` for any error messages.
- If it returns "Login failed for user '<user>'. Reason: Not associated with a trusted SQL Server connection." then check the security authentication mode on the SQL Server instance. Security authentication mode in SQL Server needs to be set to “SQL Server only” or “SQL Server and Windows”. If it is set to “Windows only”, the ACP user login will fail.

## 7.8 MySQL

To test connectivity to a remote MySQL Server instance, login to the Linux command prompt as user `acp` and perform the following checks:

```
mysql -h <servername> -u <username> -p
```

- The user name and password are in `/home/acp/acp/conf/mysql.conf`.
- If it times out, a firewall blocks traffic to the remote host and port 3306.
- If the login is successful, enter the following query:  
`show databases;`  
This should return one or more table rows. If an error is returned then the user has insufficient privileges. Recheck the configuration steps in paragraph 5.2.5.





## **7.9 WindowsWMI**

To test connectivity to a remote Windows server having the NRPE service installed, login to the Linux command prompt as user acp and perform the following checks:

```
check_nrpe -H <hostname> -p 5666 -t 30 -c check_uptime
```

- If it times out, a firewall blocks traffic to the remote host and port 5666.
- If it is successful, it should display a number of seconds indicating the uptime of the server.

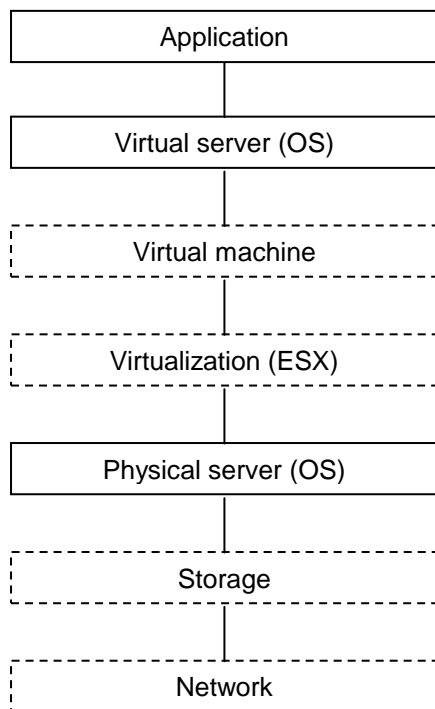


## 8 Managing infra chains

### 8.1 Layers

As from ACP v2.2, it is possible to define logical relations between infrastructural objects, to define a chain of objects that make up an application or a service. In ACP terminology this is called an infra chain. An infra chain is layered and consists of an application layer and a server layer:

The layered approach allows for definition of more layers. Theoretically, the following layers can be identified:



**Figure 5 Layers in the infrastructure**

In the current ACP version only the application layer and the physical/virtual server (OS) layers are implemented. ACP knows no distinction between a physical OS and a virtual OS.

Layers are defined in the MySQL database table “chain\_type”.

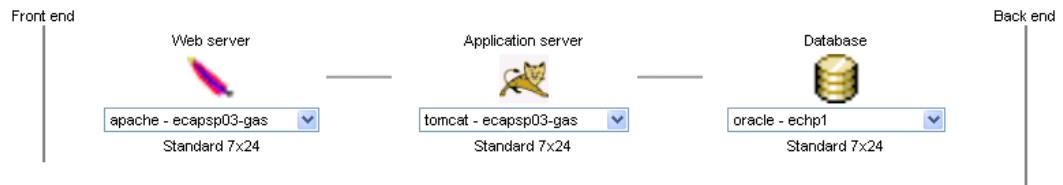


## 8.2 Tiers

In the current ACP version, each layer consists of 3 tiers.

Not every tier in a layer needs to be defined. For example, if a web server and an application server run on the same host, this host only needs to be defined once in the server layer.

### Application layer



### Server layer



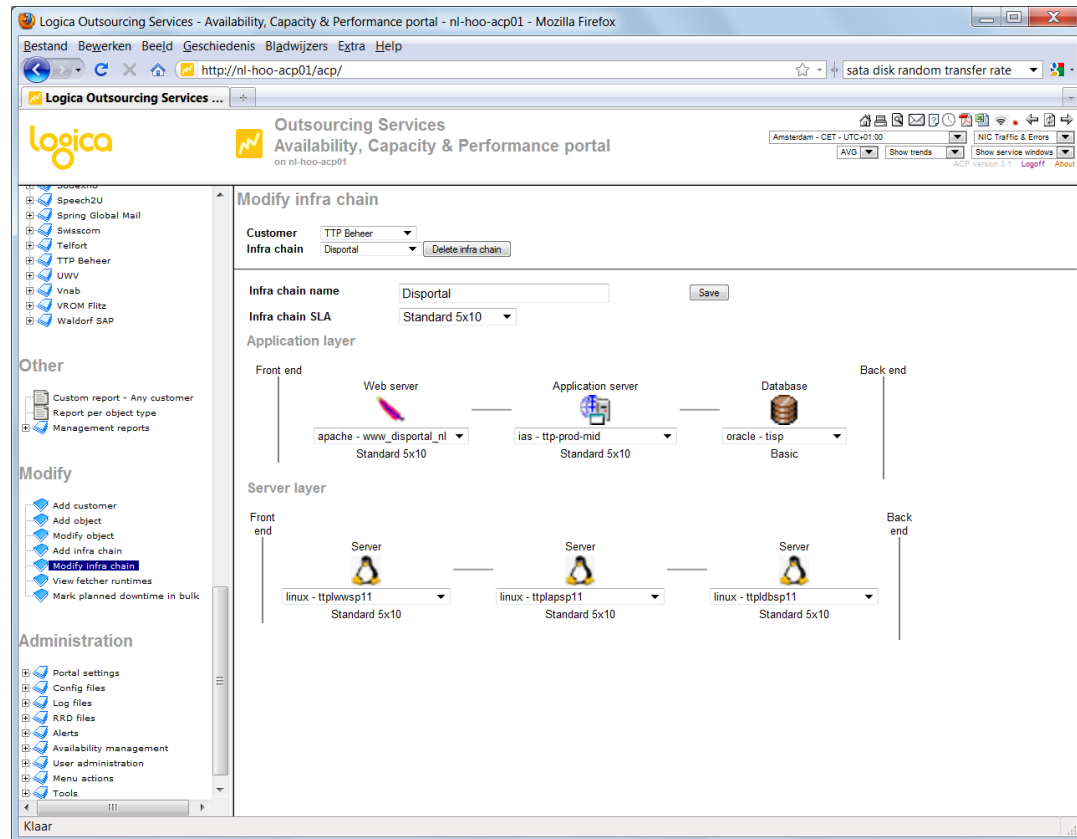
**Figure 6** Example of an infra chain

Every tier in a specific layer is suited for a limited number of object types. For example the first tier of the application layer (“Web server”) may only contain objects of type Apache or IIS. This is defined in the MySQL database table “chain\_subtype”.



## 8.3 Adding or modifying an infra chain

When adding or modifying an infra chain, the ACP super user or administrator needs to have specific knowledge of the customer estate to define object relations. ACP only knows individual objects. Logical relations between objects present in ACP can be defined in the “Add Infra Chain” and “Modify Infra Chain” screens.



**Figure 7 The Modify Infra Chain screen**

All objects in a chain should typically have the same contract and service windows, however in practice this may not be the case. In the example above, the Linux server has a different contract than the web server and database it serves. In order to calculate availability percentages for the infra chain as a whole, a contract needs to be selected on infra chain level.

Once an infra chain is saved, it will become available in the Customer overview screen.



## 9 Trend breach detection and alerting

### 9.1 Introduction

Traditional event based alerting used by monitoring frameworks raises an alert when a metric read from a monitored object exceeds one or more predefined thresholds. For example, commonly used thresholds for file systems or logical volumes are 80%, 90% and 100% usage.

Threshold based alerts are and will always be essential for detecting possible issues that require urgent or immediate attention from a technical specialist.

When a threshold based alert is raised, the threshold value represents the metric value of that specific moment in time. However, the alert does not provide any information about:

- The growth rate of the metric that led to the threshold being exceeded.  
When the metric's growth rate is low, then the alert may not require urgent attention (false positive). When the metric's growth rate is high, the issue may lead to an outage affecting the service before the reaction time defined in the SLA has passed (incident).
- Whether the measured value fits the normal behavior of the metric.  
Some metrics have recurring behavioral patterns (seasonal patterns) that may adversely trigger alerts. For example, if a certain metric fluctuates between 40% and 85% during each day as a result of normal application usage, a value of 80% as defined by the threshold may not represent an exception (false positive). From an opposite perspective, trend breaches whose values never reach a predefined threshold, may impact the service but will never result in a threshold based alert (undetected incident).

Trend breach detection and alerting is based on tracking a metric's behavior over a certain period of time and predicting its next value within a range of values (confidence band). The algorithm accounts for both linear trends and seasonal variations in the metric's value.

This provides a mechanism for alerting on conditions that can't be detected using traditional threshold based alerting. Trend breach detection can identify potential issues well before a predefined threshold is reached, which gives technical specialists more time to resolve the issue before it will adversely affect the service.

### 9.2 Holt-Winters aberrant behavior detection

#### 9.2.1 The algorithm

ACP uses the Holt-Winters time series forecasting algorithm for trend breach detection (also referred to as aberrant behavior detection), which is built in in RRDTTool.

Aberrant behavior detection is decomposed into three pieces, each building on its predecessor:

1. An algorithm for predicting the values of a time series one time step into the future.
2. A measure of deviation between the predicted values and the observed values.
3. A mechanism to decide if and when an observed value or sequence of observed values is 'too deviant' from the predicted value(s).



The Holt-Winters Time Series Forecasting Algorithm is an online, or incremental, algorithm that adaptively predicts future observations in a time series. Its forecast is the sum of three components:

- a baseline (or intercept),
- a linear trend over time (or slope), and
- a seasonal coefficient (a periodic effect, such as a daily cycle).

There is one seasonal coefficient for each time point in the period (cycle). After a value is observed, each of these components is updated via exponential smoothing. The algorithm learns from past values and uses them to predict the future. The rate of adaptation is governed by 3 parameters: alpha (intercept), beta (slope), and gamma (seasonal). The prediction can also be viewed as a 'smoothed' value for the time series.

The measure of deviation is a seasonal weighted absolute deviation. The term 'seasonal' means deviation is measured separately for each time point in the seasonal cycle. As with Holt-Winters Forecasting, deviation is predicted using the measure computed from past values (but only at that point in the seasonal cycle). After the value is observed, the algorithm learns from the observed value via exponential smoothing. Scaling the sequence of predicted deviation values for the observed time series generates confidence bands (we usually think of the sequences as continuous lines rather than as a set of discrete points).

Aberrant behavior (a potential 'failure') is reported whenever the number of times the observed value violates the confidence bands meets or exceeds a specified threshold within a specified temporal window (i.e. 6 violations during the past 90 minutes with a value observed every 10 minutes).

More technical information about the Holt-Winters forecasting algorithm can be found at:

- [http://www.usenix.org/events/lisa00/full\\_papers/brutlag/brutlag\\_html/](http://www.usenix.org/events/lisa00/full_papers/brutlag/brutlag_html/)
- [http://cricket.sourceforge.net/aberrant/rrd\\_hw.htm](http://cricket.sourceforge.net/aberrant/rrd_hw.htm)
- [http://www.it.iitb.ac.in/~praj/acads/seminar/04329008\\_ExponentialSmoothing.pdf](http://www.it.iitb.ac.in/~praj/acads/seminar/04329008_ExponentialSmoothing.pdf)
- <http://oss.oetiker.ch/rrdtool/doc/rrdcreate.en.html>



## 9.2.2 Implementation in ACP

Enabling trend breach detection on an object in ACP can only be done upon object creation. Existing objects can not be converted to enable trend breach detection. This is due to the RRDTTool implementation of the Holt-Winters forecasting algorithm, on which ACP relies.

If an object is created with Holt-Winters forecasting enabled, five Round Robin Archives (RRAs) will be added to some of its RRD files:

- **HWPREDICT**: an array of forecasts computed by the Holt-Winters algorithm, one for each Primary Data Point (PDP).
- **SEASONAL**: an array of seasonal coefficients with length equal to the seasonal period. For each PDP, the seasonal coefficient that matches the index in the seasonal cycle is updated.
- **DEVPREDICT**: an array of deviation predictions. Essentially, **DEVPREDICT** copies values from the **DEVSEASONAL** array to preserve a history; it does no processing of its own.
- **DEVSEASONAL**: an array of seasonal deviations. For each PDP, the seasonal deviation that matches the index in the seasonal cycle is updated.
- **FAILURES**: an array of boolean indicators, a 1 indicating a failure. The Consolidated Data Point (CDP) buffer stores each value within the window. Each update removes the oldest value from this buffer and inserts the new observation. On each update, the number of violations is recomputed. The maximum window length enforced by this buffer is 28 time points.

Enabling trend breach detection and alerting on an object has the following consequences:

- Some of the object's RRD files will be larger than normal as they will contain 5 extra RRAs compared to objects that do not have trend breach detection enabled.
- Updates to RRD files of the object will require more CPU and I/O because of the extra RRAs and because additional processing has to be done for alerting.
- Alerts will be sent out to either an email address or to MultiCenter. This requires staff to handle alerts and react upon alerts.

### **IN CONCLUSION: trend breach detection and alerting needs to be used sparingly!**

ACP provides a number of facilities to minimize additional resource usage and minimize the number of alerts sent out:

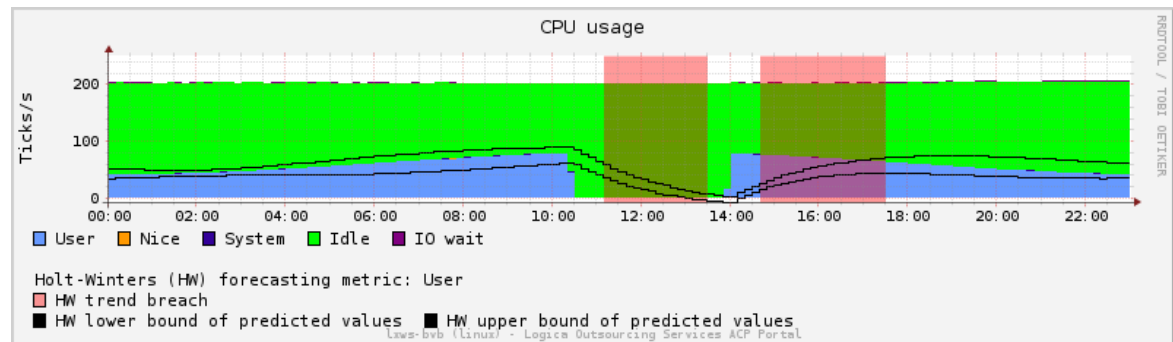
- Trend breach detection can be enabled and disabled for the entire ACP portal. By default it is enabled.
- Trend breach detection can be enabled and disabled per object. By default it is enabled.
- Alerting can be enabled and disabled per object. By default it is disabled.
- Trend breach detection is only performed on a very limited number of metrics for each object type. For example, for OS-type objects it is only performed for CPU usage, memory and logical volumes/file systems. Refer to the ACP online help to find the specific metrics used per object type.
- The first alert for an object will be sent if it occurs at least 3 seasonal periods after object creation (two periods before the confidence band appears and one period to get a stable seasonal prediction confidence band).
- An alert for an object will only be sent if it occurs at least one seasonal period after the previous alert.



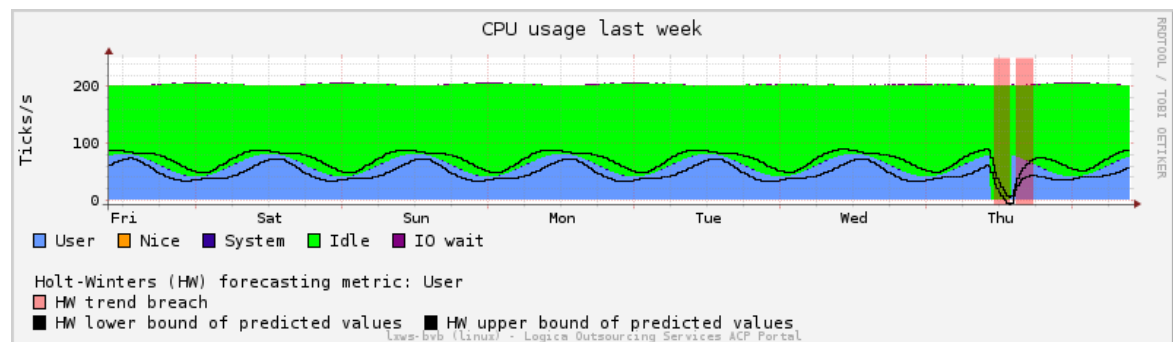
### 9.2.3 Displaying trend breaches

Since ACP v3.0, the trends dropdown box in the title frame contains an option named “Trend breaches”. If this option is selected, the content frame reloads and displays the confidence band and possible trend breaches in graphs for RRD files that have Holt-Winters forecasting enabled.

Graphs for RRD files that do not have Holt-Winters forecasting enabled will be displayed as usual without Holt-Winters information.



An ACP graph showing the Holt-Winters forecasting for CPU usage of a Linux server. The black lines represent the confidence band. The pink areas represent detected trend breaches for the downward and the upward trend. Note that graphs will only display historical confidence bands and historical trend breaches.



The same graph, displayed for a longer period. This shows the daily seasonal behavior of the CPU usage of this specific Linux server.

**Figure 8 Trend breach examples**

The graph legend now contains the following additional information:

- The forecasting metric  
ACP only displays the confidence band and trend breaches for one, distinctive data source (metric) in a graph. ACP can not be configured to use another data source.
- HW lower bound of predicted values  
This is the lower boundary of the confidence band. ACP always uses symmetrical confidence bands.
- HW upper bound of predicted values  
This is the upper boundary of the confidence band.





## 9.2.4 Holt-Winters parameters

The Holt-Winters forecasting algorithm is a quite complex algorithm that needs tuning before a trend breach can be confidently identified, thereby minimizing false positives.

ACP uses a generic set of parameter default values that will suffice for most types of metric. These will only fire an alert when a significant trend breach occurs (not too small a variation, not too late, not too soon). The confidence band adapts to the variation of the metric and as not all metric types behave in a similar way (compare disk usage against CPU usage), the default parameters chosen for the algorithm may need adapting occasionally.

Tuning Holt-Winters parameters can be an arduous task, which is the reason why it is hidden from ACP read only and super users. Holt Winters parameters can only be set or changed by ACP administrators.

These are the Holt-Winters parameters used by ACP:

Parameter	Description	Default value
Seasonal period	The number of measurements (primary data points) in a seasonal cycle. A primary data point in ACP is fixed to 600 seconds (10 minutes). This parameter can only be set upon object creation.	1008 (7 days)
Prediction retention period	The number of predictions (primary data points) to store in the RRD file before wrap around. Must be larger than the seasonal period. This parameter can only be set upon object creation.	4464 (31 days)
Alpha	Intercept adaptation parameter for the Holt-Winters forecasting algorithm. This parameter must be between 0 and 1. A larger value means the intercept adapts faster to more recent measurements.	0.1
Beta	Slope adaptation parameter for the Holt-Winters forecasting algorithm. This parameter must be between 0 and 1. A larger value means the slope adapts faster to more recent measurements.	0.03
Gamma	Seasonal coefficient adaptation parameter for the SEASONAL RRA of the Holt-Winters forecasting algorithm. This parameter must be between 0 and 1.	0.1
Gamma deviation	Seasonal deviation adaptation parameter for the DEVSEASONAL RRA of the Holt-Winters forecasting algorithm. This parameter must be between 0 and 1.	0.1
Threshold	Number of confidence bound violations that constitute a failure for purposes of raising an alert. This must be an integer less than or equal to the window length. Setting this option will reset the count of violations to 0.	6 (1 hour)
Window size	Number of time points in the temporal window for determining failures. This must be an integer greater than or equal to the threshold and less than or equal to 28. Setting this option will reset the count of violations to 0.	9 (1 hour 30 mins)
Deltapos	Deviation scaling factor for the upper bound of the confidence band used internally to calculate violations for raising alerts.	5
Deltaneg	Deviation scaling factor for the lower bound of the confidence band used internally to calculate violations for raising alerts.	5



### 9.3 Setting portal wide defaults

Holt-Winters default parameter values will be applied to all newly created objects for which trend breach detection is enabled. These default parameters can be set under Administration menu → Portal settings → Holt-Winters settings.

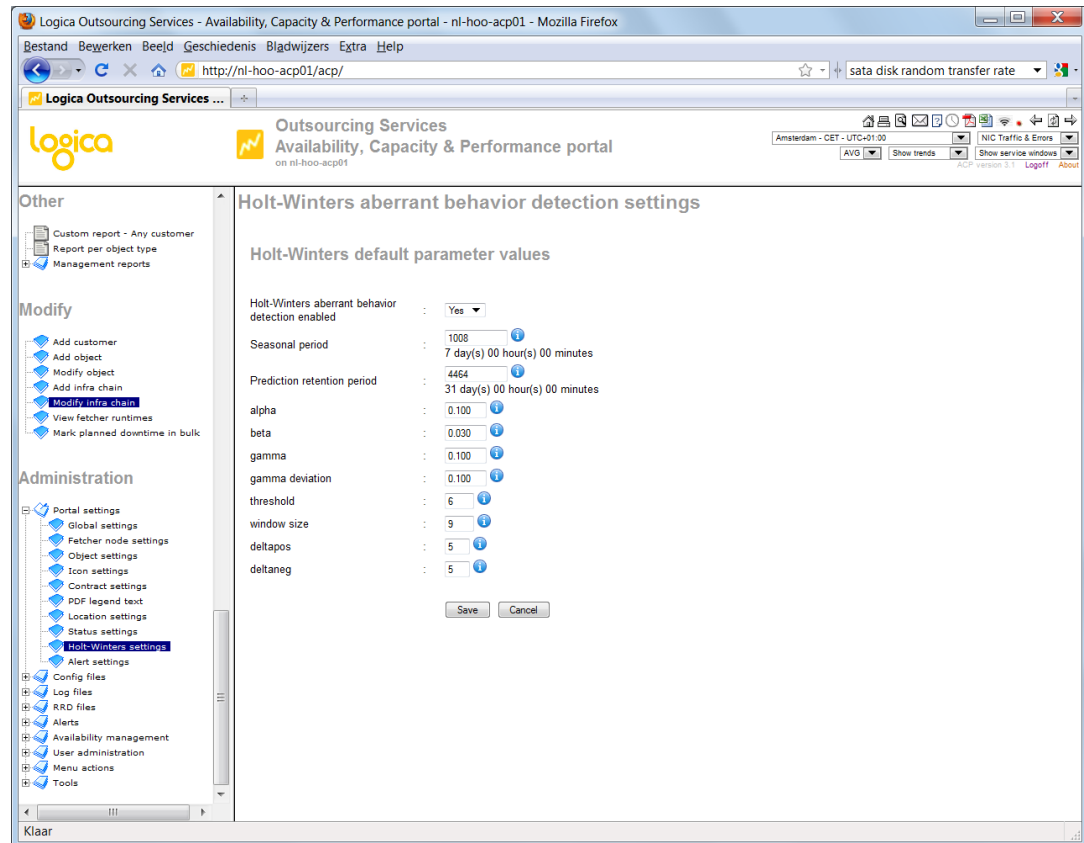


Figure 9 The Holt-Winters settings screen

Besides setting default values for Holt-Winters parameters (which are explained in paragraph 9.2.4) this screen allows to enable or disable Holt-Winters aberrant behavior detection ACP portal wide. By default, it is enabled.

If Holt-Winters aberrant behavior detection is enabled, the Add object screen will show an additional section for specifying object-specific Holt-Winters parameters. This is explained in paragraph 9.4.



## 9.4 Adding an object

If Holt-Winters aberrant behavior detection is enabled on portal level, the Add object screen will show an additional section for specifying object-specific Holt-Winters parameters:

**Trend breach detection**

Enable trend breach detection ☒

Send alert upon trend breach detection ☐

Seasonal period ⓘ 7 days ▼

Prediction retention period ⓘ 31 days ▼

This is your only chance to enable trend breach detection for the object. The checkbox “Enable trend breach detection” is enabled by default. If you uncheck this checkbox and create the object, you can never enable it again unless you delete and recreate the object. This will remove all history for the object. If you create an object with trend breach detection enabled, you can disable and enable it afterwards. The same applies to sending alerts upon trend breach detection, which is disabled by default.

A number of predefined values are present for setting the seasonal and prediction retention period:

Season	Prediction retention
1 day	7 days
7 days (default)	31 days (default)
14 days	62 days
31 days	92 days

If a custom period is chosen for the seasonal period or prediction retention period, this needs to be specified in primary data points (one primary data point equals 600 seconds). Once you change the value of either of these fields, ACP will calculate its duration in days, hours and minutes.

**Trend breach detection**

Enable trend breach detection ☒

Send alert upon trend breach detection ☐

Seasonal period ⓘ Custom period ▼ 72 0 day(s) 12 hour(s) 00 minutes

Prediction retention period ⓘ Custom period ▼ 288 2 day(s) 00 hour(s) 00 minutes

Because the seasonal period can only be set upon object creation, you would need to have an idea of its recurring behavior patterns before any ACP history is collected for the object. This is not always the case. Use the following guidelines for setting the seasonal period and prediction retention period:

- The default seasonal period is 7 days, which accounts for different metric behavior during weekends and as a result will not lead to false alerts being raised during weekends. If the behavior of the object will look more or less the same on each day (including weekends), a seasonal period of 1 day would suffice.
- If an object’s behavior is expected to show higher usage as a result of bi-weekly or monthly batch runs, consider setting the seasonal period to 14 days or 31 days.
- The prediction retention period is needed for calculating new predicted values and must therefore be long enough to get reliable predictions. It must be longer than the seasonal period. Advice is to keep it at 3 times the value of the seasonal period as a minimum.



## 9.5 Modifying an object

If Holt-Winters aberrant behavior detection is enabled on portal level and is enabled for the object, the Modify object screen will show an additional section for modifying object-specific Holt-Winters parameters:

---

**Trend breach detection**

Enable trend breach detection: ☒

Send alert upon trend breach detection: ☒

---

Note that the seasonal period and prediction retention period fields are now absent.

## 9.6 The Holt-Winters parameter file for an object

Holt-Winters settings for an object are stored in a separate file with the following name:

`<customername>_<objectname>_<objecttype>.hw`

The file is located in the same directory as the object's RRD files:

`/home/acp/acp/rrd/<objecttype>/<customername>/`

If the fetcher has never run yet, the file is located in `/home/acp/acp/rrd`, however the first fetcher run will move it to the correct directory as show above.

The contents of the Holt-Winters parameter file are:

```
TSADDED=1275997552
HWENABLE=1
HWALERT=1
HWSEASON=1008
HWWRAP=4464
HWALPHA=0.1
HWBETA=0.03
HWGAMMA=0.1
HWGAMMADEV=0.1
HWTHRESHOLD=6
HWWINDOWSIZE=9
HWDELTAPOS=5
HWDELTANEG=5
```

There is no need to change this file manually. It will be updated automatically as a result of changes in the Modify Object screen (paragraph 9.5) or the "Change Holt-Winters parameters in RRD file" screen (paragraph 9.7).

TSADDED is the Unix timestamp (seconds since Unix epoch) of the time of object creation.



## 9.7 Changing Holt-Winters parameters of an existing RRD file

Holt-Winters parameters of existing RRD files can be changed using the Administration menu → RRD files → Change Holt-Winters parameters:

### Change Holt-Winters parameters in RRD file

Monday 06 September 2010 10:27

Apply filter

Customer

Object type

Object

Search string

#### Select RRD file

test\_10.16.74.32\_linux\_cpu.rrd

test\_10.16.74.32\_linux\_fs\_-.rrd

test\_10.16.74.32\_linux\_fs\_-boot.rrd

test\_10.16.74.32\_linux\_load.rrd

test\_10.16.74.32\_linux\_mem.rrd

>>

#### Holt-Winters parameters

RRD file: **test\_10.16.74.32\_linux\_cpu.rrd**

Seasonal period 7d 00:00

Predictions stored for 31d 00:00

Deltapos

Deltaneg

Failure threshold

Window size

Alpha

Beta

Gamma

Gamma deviation

Reset datasource

**Figure 10 Change Holt-Winters parameters in an RRD file**

If there is no obvious reason for changing these parameters, leave them as they are because changing them may lead to unpredictable behavior and invalid alerts being sent out.

If you do want to change parameters, consider the following:

- Alpha, beta and gamma allow you to define how fast the confidence band adapts to changes in the metric. A larger value means it adapts faster to more recent measurements. The faster it adapts, the less likely it is that it will detect a trend breach. Alpha and beta are for linear variations and gamma for seasonal variations.
- Window size and threshold allow you to define how soon observed values that lie outside the confidence band will be regarded as a trend breach. By default, ACP detects a trend breach when 6 out of 9 observed values (90 minutes) lie outside the confidence band.
- The Holt-Winters algorithm works quite well when there is some level of variance in the metric's baseline. In case of non-changing metrics (for example a non growing file system) the confidence band narrows, so that small changes in the metric will be regarded as trend breaches (false positive). In that case you could tune the confidence band to adapt as fast as possible.



Next to changing Holt-Winters parameters, this screen also allows you to reset the aberrant behavior detection algorithm for a specified data source in the RRD file; that is, forget all it has learnt so far.

Specifically, for the HWPREDICT or MHPREDICT RRA, it sets the intercept (alpha) and slope (beta) coefficients to unknown. For the SEASONAL RRA, it sets all seasonal coefficients (gamma and gamma deviation) to unknown. For the DEVSEASONAL RRA, it sets all seasonal deviation coefficients to unknown. For the FAILURES RRA, it erases the violation history. Note that reset does not erase past predictions (the values of the HWPREDICT or MHPREDICT RRA), predicted deviations (the values of the DEVPREDICT RRA), or failure history (the values of the FAILURES RRA).

Use of this tuning option is advised when the behavior of the data source time series changes in a drastic and permanent manner.

More technical information on how to tune RRD files can be found at <http://oss.oetiker.ch/rrdtool/doc/rrdtune.en.html>.



## 9.8 Trend breach alert settings

Alerting methods can be changed using the Administration menu → Portal settings → Alert settings.  
ACP currently supports two methods for alerting:

- Email (inactive by default)  
This method requires an email relay gateway that is configured to accept email from the ACP server. Also, the exim4 daemon on the ACP server needs to be configured to send out emails through this relay. Parameter 1 is the email address to send emails to.
- Multicenter (active by default)  
This method requires a MultiCenter agent to be installed and running on the ACP server. In MultiCenter, an MC logscan definition has to be setup on the ACP server that regularly scans the ACP log and transfers the alerts to the MEC. Parameter 1 is the log file location and name. Parameter 2 is the MC severity level to be used.

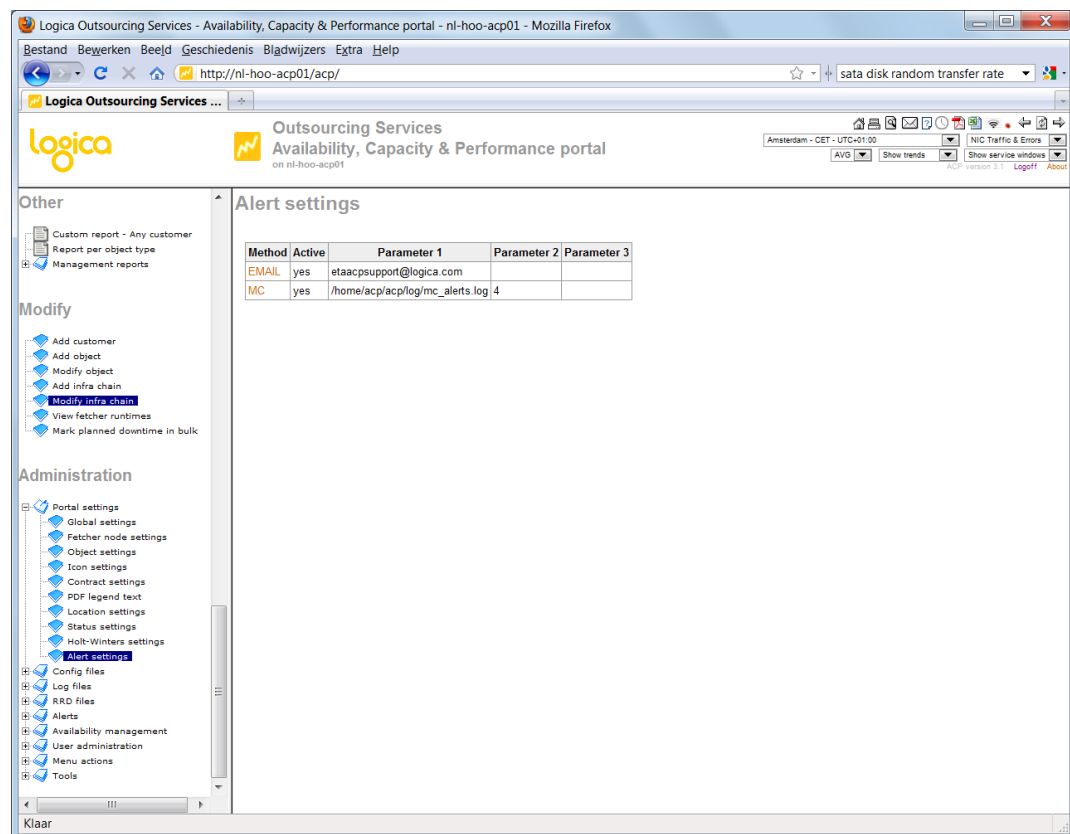


Figure 11 Trend breach alert settings

The Multicenter logfile is located in `/home/acp/acp/log/mc_alerts.log`.

The record format of an MC alert is:

```
TIMESTAMP=<yyyymmdd-hh24miss>;ALERTHOST=<hostname>;SEVERITY=4;CUSTOMER=<name>;OBJECTTYPE=<type>;OBJECT=<name>;MESSAGE=<text>
```

Example:

```
TIMESTAMP=20100812-150000;ALERTHOST=localhost;SEVERITY=4;CUSTOMER=test;OBJECTTYPE=linux;OBJECT=localhost;MESSAGE=Trend breach detected in graph "cpu" for metrics(s) "user"
```

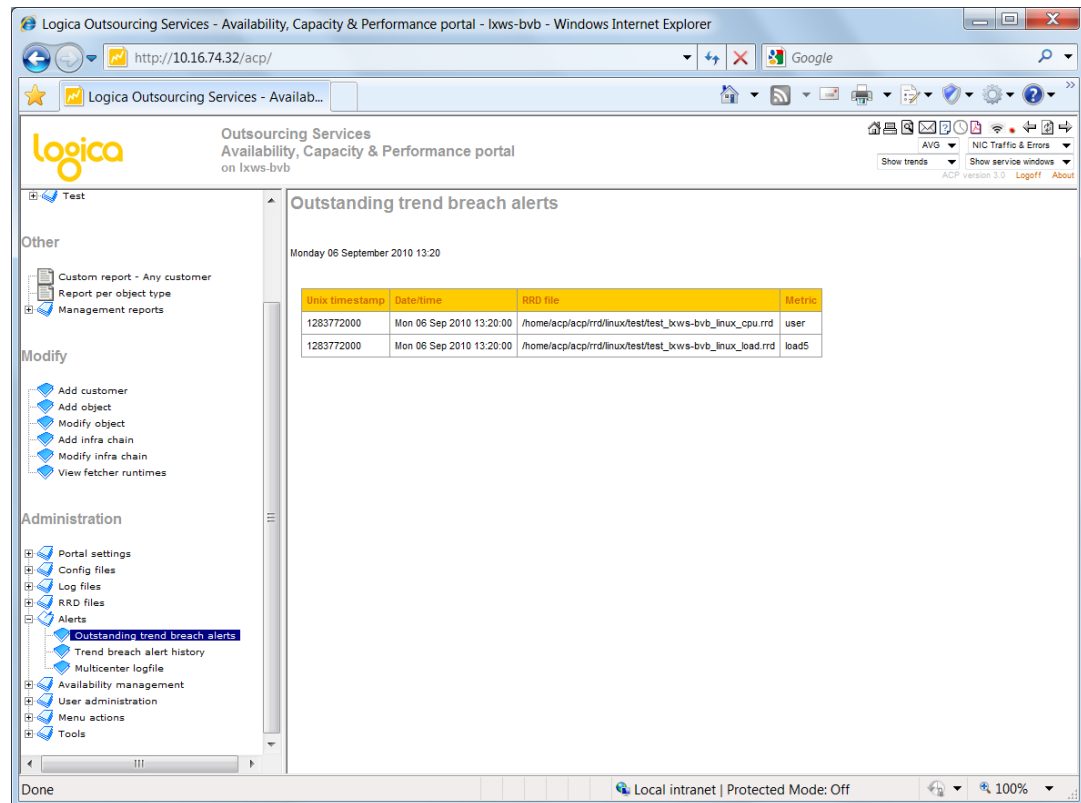
If the object is registered with its hostname in ACP, ALERTHOST will hold the hostname on which the alert occurred. Otherwise it will contain the ACP server name.



## 9.9 Viewing trend breach alerts

### 9.9.1 Outstanding trend breach alerts

Outstanding alerts can be found under the Administration menu → Alerts → Outstanding trend breach alerts:



**Figure 12 Outstanding trend breach alerts**

This screen shows active trend breaches (status equals 1) for which an alert has been sent out, irrespective of the alerting method.

The table has the following columns:

- Timestamp in seconds since Unix epoch (January 1<sup>st</sup>, 1970) of when the alert was raised.
- Timestamp in calendar date/time of when the alert was raised.
- The location and name of the RRD file. This contains the customer name, object type, object name and, if applicable, the item name.
- Metric: the data source name in the RRD file for which a trend breach was detected. Refer to the online help for a description of the metric.

If the trend breach is cleared, it will disappear from this screen and will become visible in the trend breach history.

NOTE: alerts on objects for which alerting is disabled will not appear in this screen.





## 9.9.2 Trend breach alert history

The trend breach alert history can be found under the Administration menu → Alerts → Trend breach alert history:

Logica Outsourcing Services - Availability, Capacity & Performance portal - lxws-bvb - Windows Internet Explorer

http://10.16.74.32/acp/

Logica Outsourcing Services - Availab...

logica Outsourcing Services  
Availability, Capacity & Performance portal  
on lxws-bvb

Test

Other

- Custom report - Any customer
- Report per object type
- Management reports

Modify

- Add customer
- Add object
- Modify object
- Add infra chain
- Modify infra chain
- View fetcher runtimes

Administration

- Portal settings
- Config files
- Log files
- RRD files
- Alerts
  - Outstanding trend breach alerts
  - Trend breach alert history**
  - Multicenter logfile
- Availability management
- User administration
- Menu actions
- Tools

Trend breach alert history

Monday 06 September 2010 13:20

Delete alert history older than  
30 days

Delete

Unix timestamp	Date/time	RRD file	Metric	Status
1283772000	Mon 06 Sep 2010 13:20:00	/home/acpl/acplrrd/linux/test/test_kws-bvb_linux_cpu.rrd	user	1
1283772000	Mon 06 Sep 2010 13:20:00	/home/acpl/acplrrd/linux/test/test_kws-bvb_linux_load.rrd	load5	1
1283530200	Fri 03 Sep 2010 18:10:00	/home/acpl/acplrrd/linux/test/test_kws-bvb_linux_mem.rrd	memrealused	0
1283526000	Fri 03 Sep 2010 17:00:00	/home/acpl/acplrrd/linux/test/test_kws-bvb_linux_mem.rrd	memrealused	1
1283515800	Fri 03 Sep 2010 14:10:00	/home/acpl/acplrrd/linux/test/test_kws-bvb_linux_mem.rrd	memrealused	0
1283513400	Fri 03 Sep 2010 13:30:00	/home/acpl/acplrrd/linux/test/test_kws-bvb_linux_mem.rrd	memrealused	1
1283397000	Thu 02 Sep 2010 05:10:00	/home/acpl/acplrrd/linux/test/test_kws-bvb_linux_fs_-.rrd	fsuse	0
1283392800	Thu 02 Sep 2010 04:00:00	/home/acpl/acplrrd/linux/test/test_kws-bvb_linux_cpu.rrd	fsuse	1
1283166000	Mon 30 Aug 2010 13:00:00	/home/acpl/acplrrd/linux/test/test_kws-bvb_linux_cpu.rrd	system	0
1283164800	Mon 30 Aug 2010 12:40:00	/home/acpl/acplrrd/linux/test/test_kws-bvb_linux_cpu.rrd	system	1
1283161800	Mon 30 Aug 2010 11:50:00	/home/acpl/acplrrd/linux/test/test_kws-bvb_linux_cpu.rrd	system	0
1283161200	Mon 30 Aug 2010 11:40:00	/home/acpl/acplrrd/linux/test/test_kws-bvb_linux_cpu.rrd	system	1
1282665000	Tue 24 Aug 2010 17:50:00	/home/acpl/acplrrd/linux/test/test_kws-bvb_linux_cpu.rrd	system	0
1282660800	Tue 24 Aug 2010 16:40:00	/home/acpl/acplrrd/linux/test/test_kws-bvb_linux_cpu.rrd	system	1
1282651800	Tue 24 Aug 2010 14:10:00	/home/acpl/acplrrd/linux/test/test_kws-bvb_linux_cpu.rrd	user	0
1282648200	Tue 24 Aug 2010 13:10:00	/home/acpl/acplrrd/linux/test/test_kws-bvb_linux_cpu.rrd	system	0

Done

Local intranet | Protected Mode: Off

100%

Figure 13 Trend breach alert history

This screen shows a history of when trends breach alerts were raised or cleared. In comparison to the Outstanding trend breach alerts screen, it has an additional Status column. A status of 1 means an alert was raised and a status of 0 means it was cleared.

NOTE: alerts on objects for which alerting is disabled will not appear in this screen.



### 9.9.3 Multicenter logfile

The contents of the Multicenter logfile can be found under the Administration menu → Alerts → Multicenter logfile:

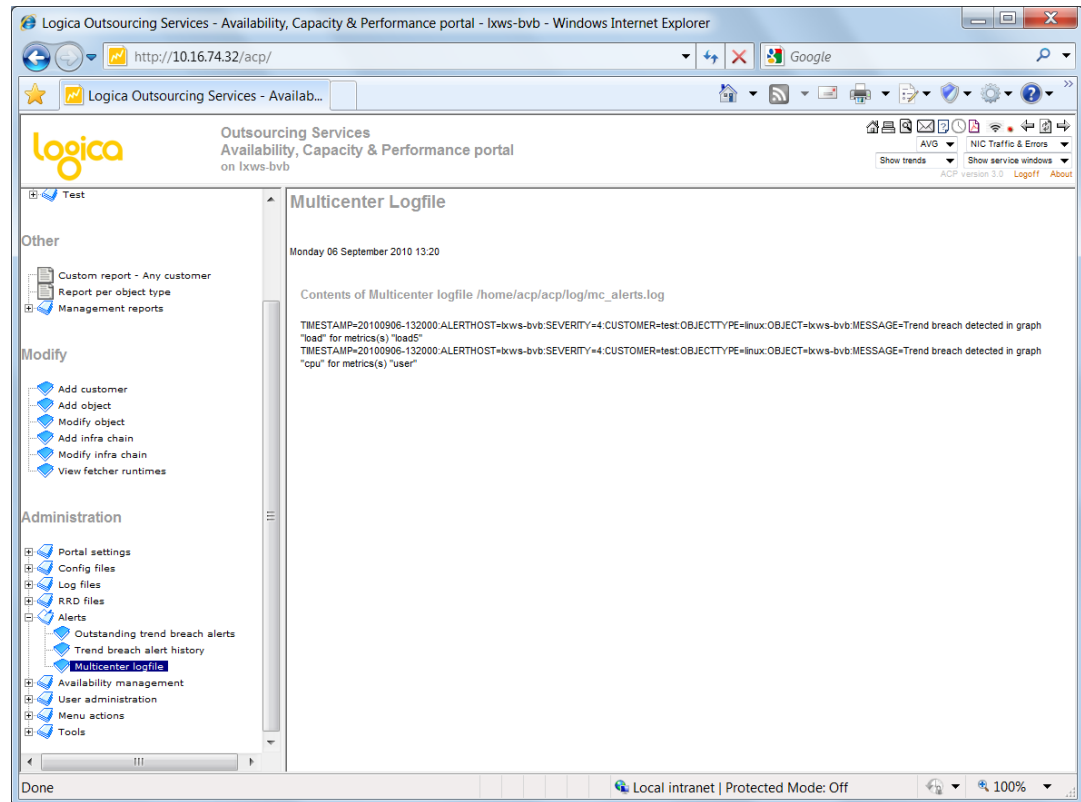


Figure 14 Contents of the Multicenter logfile

The Multicenter logfile only contains records of raised trend breach alerts. Clearance of an alert is not written to the Multicenter logfile.

NOTE: alerts on objects for which alerting is disabled will not appear in the Multicenter logfile.



## 10 Administration menu

### 10.1 Portal settings

#### 10.1.1 Global settings

The global settings page contains the following settings:

- ACP base directory: the directory in which the ACP application is installed on the Linux server. Defaults to /home/acp/acp
- RRD files directory: the directory in which the RRD files are stored by the fetchers. Defaults to /home/acp/acp/rrd
- RRD files backup directory: the directory to which RRD files are moved when an object is deleted. Defaults to /home/acp/acp/rrdbackup
- Colors for headings and tables
- A drop down box to select whether the “Select tree view” drop down box will be shown in the Customers menu.
- Uptime threshold for SLA reports: this is the default value displayed for the checkbox “Only show objects with uptime above/below n%” in SLA reports selection boxes in the Customer Overview screen and Customer-Object Overview screen. Defaults to 90.
- Default password for new users. Defaults to welcome12.
- The number of days of inactivity after which a user account will be locked.
- A system message that will be displayed in the content frame after a user has logged on to ACP.

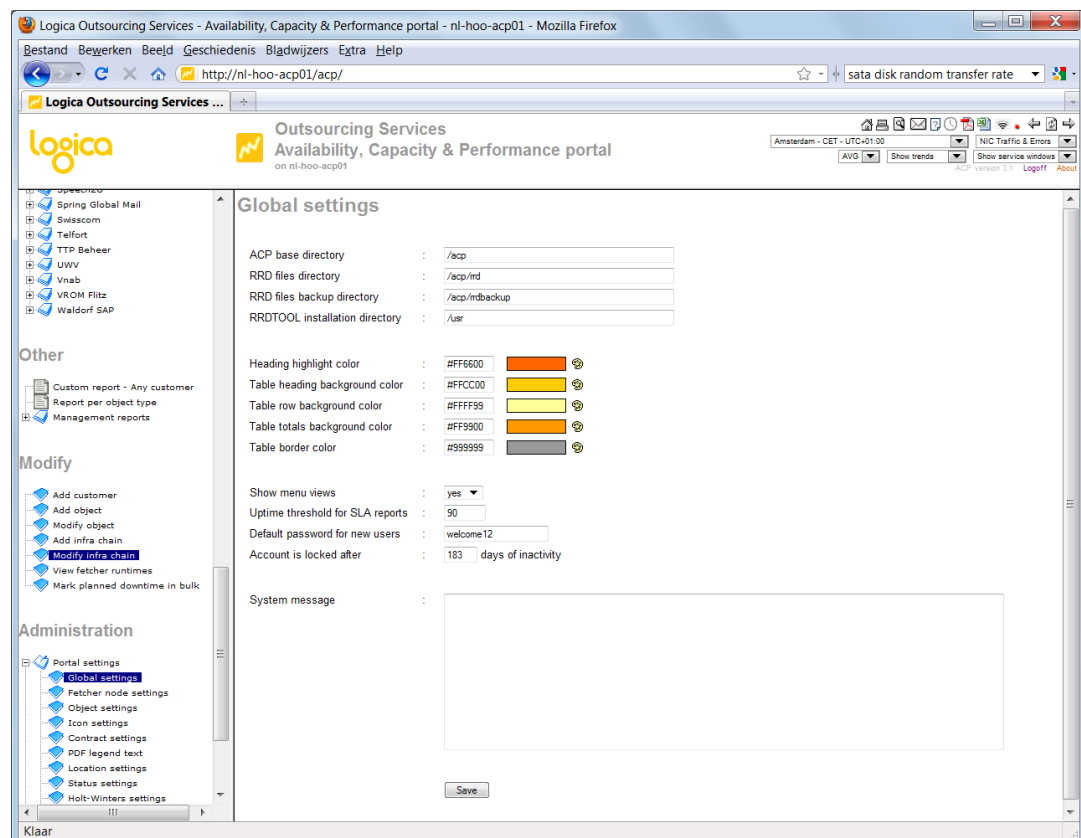


Figure 15 The Global settings screen



### 10.1.2 Fetcher node settings

The fetcher node settings screen is only used in a shared storage infrastructure, where there are multiple ACP servers for fetching. It allows enabling or disabling addition or modification of objects on each fetcher node.

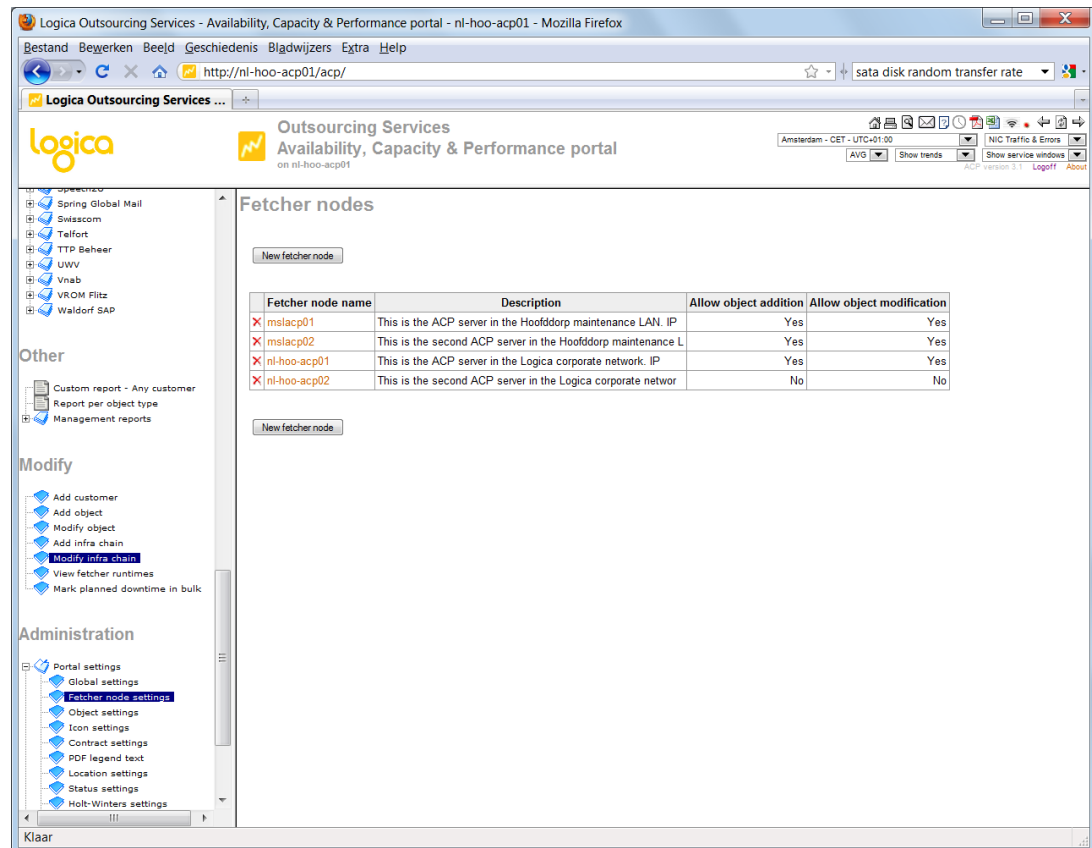


Figure 16 The Fetcher nodes screen



### 10.1.3 Object settings

In the Object settings screen object types are defined. During normal operation it is not necessary to manually add or modify object types, as this table is updated during installation and with every ACP release.

The object name is the name that appears in the tree menu as an object type below a customer and in graph reports in the title.

The fetcher name needs to be lower case and usually is similar to the object name. The fetcher name must be identical to the object type that is part of the fetcher script name. For example, the fetcher script for Oracle is called `acp_fetch_oracle.py`. In this case the object name is Oracle (initial capital) and the fetcher name is oracle (lowercase).

The object label is an indication of the type of component (server, database, firewall, etc).

The screenshot shows the 'Object types' section of the ACP portal. On the left is a navigation menu with categories like 'Other', 'Modify', and 'Administration'. The 'Object types' table lists various object types with their IDs, names, labels, fetcher names, and owners. A 'New object type' button is located above the table.

ID	Object type name	Object type label	Fetcher name	Owner
X 36	3COMSwitch		3comswitch	Telecoms
X 20	AIX	server	aix	Unix
X 5	Apache	server	apache	DBA
X 27	CatOSSwitch		catosswitch	Telecoms
X 24	Checkpoint	firewall	checkpoint	Telecoms
X 51	CheckpointVPNEdge		checkpointvpndge	Telecoms
X 52	CiscoASA	firewall	ciscoasa	Telecoms
X 53	CiscoCSS	load balancer	ciscocss	Telecoms
X 37	F5BigIP	load balancer	f5bigip	Telecoms
X 33	FDS	LDAP server	fds	DBA
X 17	Hpux	server	hpux	Unix
X 6	IAS	server	apache	DBA
X 28	IIS	server	iis	Wintel
X 31	JBoss	server	jboss	DBA
X 34	JuniperSA	SSL VPN	junipersa	Telecoms
X 22	L3switch		l3switch	Telecoms
X 15	Linux	server	linux	Unix
X 45	MSEExchange	server	msexchange	Wintel
X 3	MySQL	database	mysql	DBA
X 10	Netapp	filer	netapp	Storage
X 26	Netcache	proxy	netcache	Telecoms
X 35	NortelAlteon	SSL switch	nortelalteon	Telecoms
X 39	NortelSignallingserver		nortelsignallingserver	Telecoms
X 38	NortelSwitch		nortelswitch	Telecoms
X 14	OC4J	server	oc4j	DBA
X 1	Oracle	database	oracle	DBA

Figure 17 The Object settings screen



### 10.1.4 Icon settings

In the icon settings screen, an icon can be defined for each object type/contract combination. During normal operation it is not necessary to manually add or modify icons, as this table is updated during installation and with every ACP release. If no icon is defined for an object type, a generic book icon is displayed.

The screenshot shows the 'Icons' settings page in the Logica Outsourcing Services portal. The page has a sidebar with navigation options like 'Other', 'Modify', and 'Administration'. The main area contains a table with columns: ID, Icon, Image, Object type, and Contract. The table lists various icons for different object types and contracts, such as database, sqlserver, mysql, apache, oracle\_ias, and windows.

ID	Icon	Image	Object type	Contract
X 3	database-g.png		Oracle	Standard 7x24
X 2	database-s.png		Oracle	Standard 5x10
X 1	database-b.png		Oracle	Basic
X 143	database-s.png		Oracle	Standard 5x16
X 192	database-b.png		Oracle	Standard 5x9
X 11	sqlserver-g.gif		SQLServer	Standard 7x24
X 9	sqlserver-s.gif		SQLServer	Standard 5x10
X 10	sqlserver-b.gif		SQLServer	Basic
X 144	sqlserver-s.gif		SQLServer	Standard 5x16
X 193	sqlserver-s.gif		SQLServer	Standard 5x9
X 6	mysql-icon.gif		MySQL	Standard 7x24
X 21	mysql-icon.gif		MySQL	Standard 5x10
X 22	mysql-icon.gif		MySQL	Basic
X 145	mysql-icon.gif		MySQL	Standard 5x16
X 194	mysql-icon.gif		MySQL	Standard 5x9
X 4	apache_icon.png		Apache	Standard 7x24
X 19	apache_icon.png		Apache	Standard 5x10
X 20	apache_icon.png		Apache	Basic
X 146	apache_icon.png		Apache	Standard 5x16
X 195	apache_icon.png		Apache	Standard 5x9
X 13	oracle_ias.gif		IAS	Standard 7x24
X 29	oracle_ias.gif		IAS	Standard 5x10
X 30	oracle_ias.gif		IAS	Basic
X 147	oracle_ias.gif		IAS	Standard 5x16
X 196	oracle_ias.gif		IAS	Standard 5x9
X 7	windows-icon-g.gif		Windows	Standard 7x24

Figure 18 The Icon settings screen



### 10.1.5 Contract settings

The contract settings allows for definition of SLA contracts. As from ACP version 1.3 it is possible to define custom contracts and service windows. By default, the following contracts are defined:

Contract	Service window
Standard 7x24	24*7
Standard 5x9	Monday till Friday 08:00 – 17:00
Standard 5x10	Monday till Friday 08:00 – 18:00
Standard 5x12	Monday till Friday 07:00 – 19:00
Standard 5x16	Monday till Friday 06:00 – 22:00
Basic	Monday till Friday 08:00 – 18:00 (best endeavor support)

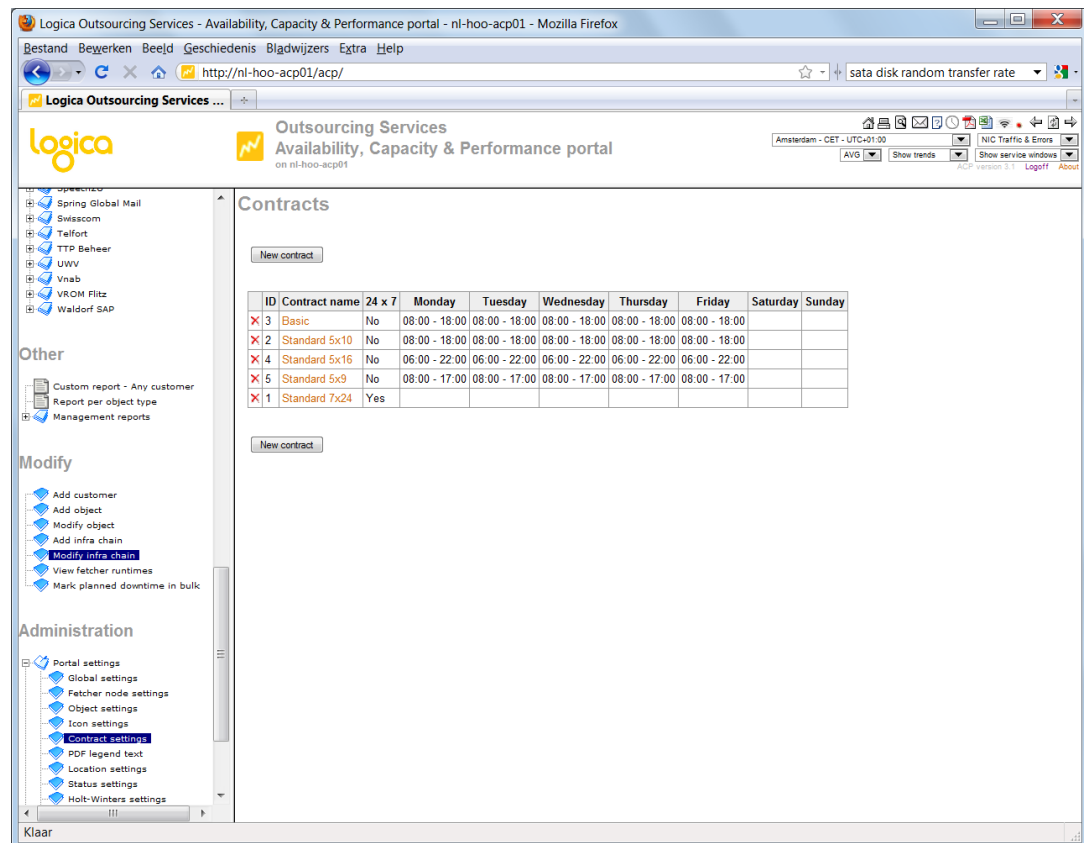


Figure 19 The Contract settings screen



For non-24x7 contracts the service window can be set for each day of the week. Only one period can be defined per day.

**Contracts**

Update contract

Contract name: Standard 5x10

☐ 24x7 ☒ Other

From To

☒ Monday 08:00 - 18:00

☒ Tuesday 08:00 - 18:00

☒ Wednesday 08:00 - 18:00

☒ Thursday 08:00 - 18:00

☒ Friday 08:00 - 18:00

☐ Saturday 00:00 - 00:00

☐ Sunday 00:00 - 00:00

ID	Contract name	24 x 7	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
3	Basic	No	08:00 - 18:00	08:00 - 18:00	08:00 - 18:00	08:00 - 18:00	08:00 - 18:00		
2	Standard 5x10	No	08:00 - 18:00	08:00 - 18:00	08:00 - 18:00	08:00 - 18:00	08:00 - 18:00		
4	Standard 5x16	No	06:00 - 22:00	06:00 - 22:00	06:00 - 22:00	06:00 - 22:00	06:00 - 22:00		
5	Standard 5x9	No	08:00 - 17:00	08:00 - 17:00	08:00 - 17:00	08:00 - 17:00	08:00 - 17:00		
1	Standard 7x24	Yes							

Figure 20 Service window definition for non-24x7 contracts

### 10.1.6 PDF legend text

The PDF legend text screen allows defining sections and text displayed in the introduction page of PDF reports generated by ACP. These are prefilled by the ACP installation, so during normal operation it is not necessary to manually edit them.

### 10.1.7 Location settings

The Location settings screen allows definition of data center locations. When an object is added, it will have an unknown location by default. For this reason the “Unknown” location can not be removed or altered.

### 10.1.8 Status settings

The Status settings screen allows definition of production statuses. When an object is added, it will have an unknown status by default. For this reason the “Unknown” status can not be removed or altered.





## 10.2 Config files

### 10.2.1 Fill object config table



**This script is only to be used in emergency situations where object configuration details for fetching are lost from the MySQL database.**

The script performs the following steps:

1. Truncate the object\_config table
2. Insert objects from the phplayersmenu table into the object\_config table
3. Read through all current config files to add additional information to the object\_config table

To be able to reconstruct the database, it is essential that all acp config files containing the original object details are available in the /home/acp/acp/conf directory.

### 10.2.2 Generate config files

This script can be used to either generate all ACP config files, or only one config file for a specific object type. The config file of an object type is regenerated when an object is added, modified or deleted, so during normal operation it is not necessary to manually regenerate it.

### 10.2.3 Import config file

This script is used for:

- Initial object imports after installation of ACP
- Batch imports of objects

The import procedure needs a text file having exactly the same structure as a normal ACP config file. The first line of the import file must contain the object type of the objects listed in the import file.

For example a Windows import file would look like:

Windows				
server1	2c	myCommString	blr	blr
server2	2c	myCommString	blr	blr
server3	2c	myCommString	blr	blr
server4	2c	myCommString	blr	blr
server5	2c	myCommString	blr	blr
server6	2c	myCommString	blr	blr
...				

An import file needs to be placed in the /home/acp/acp/conf directory and therefore should have a different name than the normal config file for that object type.

An import only imports object details in the MySQL database. After the import, the corresponding config file and the acp crontab will be regenerated.



#### 10.2.4 View config file

This script is used to view the contents of ACP config files located in the /home/acp/acp/conf directory. Some config files contain clear text passwords, so make sure these cannot be read from the screen by others.

#### 10.2.5 Regenerate ACP crontab

This script regenerates the ACP crontab. The crontab is regenerated when an object is added, modified or deleted, so during normal operation it is not necessary to manually regenerate it.

#### 10.2.6 View ACP crontab

This script shows the contents of the ACP crontab.

#### 10.2.7 View fetcher runtimes

This script is used to check if fetchers scheduled in the crontab exceed their maximum runtime of 10 minutes. The script reads all fetcher log files that correspond to the fetchers in the crontab, calculates the runtime for each fetcher run in the log file, and calculates statistics based on the number of runs.

If a fetcher run exceeds 10 minutes duration, it is displayed in red. Below is an example of the output of this script.

Object type	Fetcher name	Number of objects	Min. runtime [mm:ss]	Max. runtime [mm:ss]	Avg. runtime [mm:ss]	Last runtime [mm:ss]	Number of runs	Number of runs > 10 min	Percentage runs > 10 min
Apache	gecis	1	00:07	00:26	00:10	00:10	281	0	0 %
F5BigIP	sgnfm	1	00:08	00:50	00:16	00:12	280	0	0 %
IIS	ltube	1	00:09	00:30	00:12	00:11	280	0	0 %
JuniperSA	icube	2	00:15	02:53	00:47	00:54	280	0	0 %
Linux	gecis	2	00:04	01:26	00:22	00:24	281	0	0 %
MySQL	gecis	1	00:01	00:26	00:05	00:03	281	0	0 %
Netcache	icube	2	00:08	00:31	00:13	00:13	279	0	0 %
Oracle	logica	3	00:00	09:18	02:47	02:04	561	0	0 %
PIX	icube	7	01:32	01:35	01:33	01:33	280	0	0 %
Router	icube-nl-hoo	3	00:14	00:54	00:25	00:27	278	0	0 %
Switch	icube-nl-alk	6	01:35	10:35	04:02	02:11	274	5	2 %
Windows	logica-be	1	00:00	00:06	00:01	00:00	281	0	0 %
Windows	ltube	1	00:08	00:34	00:17	00:18	279	0	0 %



### 10.2.8 View fetcher statistics

As from ACP v2.0, all fetcher scripts also collect statistics about their own runtime behavior and write these to RRD files. For each object type, one RRD file is present. The graphs generated from this RRD file show, successful runs, unsuccessful runs and unsuccessful runs with script exceptions. This gives the ACP administrator a better overview of how fetchers run and better information for problem analysis.

### 10.2.9 Edit tnsnames.ora

This script can be used to edit the Oracle configuration file `tnsnames.ora`, which in a standard ACP installation is located in `/oracle/net/network/admin`. For each Oracle instance monitored by ACP, a connection descriptor with the following format should be added to `tnsnames.ora`. The DBA team should provide details for hostname, port and ORACLE\_SID. Usually the TNS-alias is identical to the Oracle instance name (ORACLE\_SID).

```
<TNS-alias>=
  (DESCRIPTION=
    (ADDRESS_LIST=
      (address=(protocol=tcp) (host=<hostname>) (port=<port>))
    )
    (CONNECT_DATA=(SID=<ORACLE_SID>))
  )
```



## 10.3 Log files

### 10.3.1 View general log

Fetcher scripts write error conditions to the general ACP log file,  
`/home/acp/acp/log/general.log`

The view general log script parses the general log file to show the errors in tabular format.

### 10.3.2 View fetcher log

This script is used to view the contents of all fetcher log files. Fetcher log files are located in  
`/home/acp/acp/log/<fetchertype>/<fetchername>.log`

### 10.3.3 View change history

This screen shows object additions, modifications and deletions in tabular format. The table may be sorted ascending and descending on all columns by clicking on the column heading.

It contains to additional options:

- To search for (part of) an object name
- To purge the change history older than a certain number of days.

### 10.3.4 View usage history

The usage history shows which were reports created by users. The table may be sorted ascending and descending on all columns by clicking on the column heading. It also contains an option to purge the usage history older than a certain number of days.

This report provides insight in:

- Which report types are mostly used
- Which users create the most reports (and which ones)
- Whether ACP was used at specific times (for incident analysis)



## 10.4 RRD files

### 10.4.1 Remove spikes



**This script removes data from RRD files, so it only should be used by experienced ACP administrators**

Metrics that return only increasing values are called counters. RRDTool generates graph data by subtracting subsequent counter values in the RRD file. If a counter is reset to zero (for example because the monitored object has been brought down and up again) or in case of a counter wrap (the maximum value of a 32-bits counter has been reached and therefore restarts at zero), the graph may display a spike in the data, as a negative number is interpreted as a very large positive number.

Normally the fetcher script detects and removes spikes. In some cases spikes may still show in the graph, which can be removed with the Remove spikes script.

## 10.5 Availability management

### 10.5.1 Site wide outages

Site wide outages interrupt ACP monitoring for a certain period. Interruptions in ACP monitoring may result in white space in the graphs and possible unknown time.

In the Site wide outages screen, the ACP administrator is able to enter a notice of any interruption that occurred. This includes start date/time, end date/time, a notification text and a selection of customers affected by the outage.

If a user selects a report period in which the interruption occurred, the notice will be displayed in the report notes.

### 10.5.2 Health report

For detecting issues or changes in object responsiveness, the health check report provides a table of all object defined in ACP, listing the following numbers and percentages per customer/object type:

- Total count
- Responders (100% uptime)
- Non-responders (0% uptime)
- Fluctuating (0% < uptime < 100%)
- Not monitored

### 10.5.3 Mark planned downtime in bulk

As from ACP v3.1, downtime can be marked as planned for multiple objects in one operation. This is done in the “Mark planned downtime in bulk” screen, which is only available to super users and administrators. This allows the user to select multiple objects of one type and mark all downtime that occurred during the selected period as planned downtime.

Note:

- A comment for the planned downtime is mandatory
- Only downtime that is fully enclosed by the selected period will be marked as planned. In other words, downtime that started before or ended after the selected period will not be marked as planned.



## 10.6 User administration

### 10.6.1 Add user

In the Add user screen the following options are available:

- Username: this may not contain spaces
- Administrator: checking this box makes the user an ACP administrator with access to all menus.
- Super user:: checking this box makes the user an ACP super user with access to all menus but the Administration menu.
- Customer: checking this box makes the user an customer with read only access to data of only one customer (his own data). By default, a customer has a delay time of 43200 seconds (12 hours).
- Delay time: if a number of seconds is specified in this field, the user won't see graph data in the most recent period defined by the number of seconds. So if delay time is set to 14400 seconds, the user won't be able to see graph data of the most recent 4 hours. Setting a delay time also disables displaying trend lines that extend into the future (forecasts).
- Password expires: specifies the number of days after which the user has to change his password.
- Lock account after *n* days of inactivity
- Allow login through URL: this allows a user to login directly without supplying a password. The can user can directly request an ACP page and in the process validate himself by supplying a hash key in the URL. The hash key can be obtained by clicking "Show password hash" in the Modify user screen.

**Example:**

```
http://<acpservname>/acp/bin/object_overview.php?c=sgdba&ot=Linux&o=msldb01&s  
=day&auth_user=<username>&auth_hash=e98c9200b56a15307f24984a99d5a4fd
```

- Log login attempts: by default, login attempts are logged. For frequent logins by other tools, this option can be disabled.

If none of the Administrator, Super user or Customer checkboxes are checked, the user will become a read only ACP user.

After user creation, the password is set to the default password defined in the Global settings screen (paragraph 10.1.1). By default, the default password is set to welcome12. After first logon, the user has to change his password and relogin.

### 10.6.2 Modify user

The Modify user screen has the same options as the add user screen. Additionally, this screen has two fields to change the password. By checking the Default password checkbox, the user's password will be reset to the default password defined in the Global settings screen (paragraph 10.1.1).



### **10.6.3 User privileges**

This screen is used to grant read only ACP users access to customers in the menu. By default, a read only user has no access to any customers. ACP administrators and super users are not listed in this screen.

If the checkbox “User is allowed to view all customers” is checked, no specific customer access needs to be granted anymore.

ACP users created as customers can only be granted access to data of maximum one customer.

### **10.6.4 All user report templates**

This screen lists all report templates created by users. The ACP administrator can delete a template by clicking on the red cross icon corresponding to the template. This will remove the template and all its details and schedules without notice.

As from ACP v3.1 it is possible to copy or move an existing template to another ACP user. Note that the target user must have privileges to view data of the customer the template is created on.

### **10.6.5 List ACP users online**

This screen shows the users currently logged on to ACP.

### **10.6.6 Scheduled report templates**

This screen lists all user templates scheduled for email. By clicking on the red cross icon corresponding to a schedule, the schedule will be removed without notice. The template itself will be retained.

### **10.6.7 View login history**

This screen shows successful and unsuccessful logons to ACP in tabular format. The table may be sorted ascending and descending on all columns by clicking on the column heading.

It also contains an option to purge the login history older than a certain number of days.

## **10.7 Menu actions**

### **10.7.1 Delete customer**

This screen is used to drop customers from ACP. When a customer is dropped:

- All customer and object data will be deleted from the MySQL database
- All RRD files will be backed up

## **10.8 Tools**

### **10.8.1 SSH terminal**

This screen contains a Java applet SSH terminal to the ACP server. This allows an ACP administrator to logon to the Linux server via the web interface. In order for this to work the Java plugin must be installed and enabled in the browser.