



# **How to integrate iTop and Nagios for incident management**

## How to automate the creation of incident tickets in iTop

This section describes how to configure Nagios to create automatically incident tickets in iTop. For this purpose Combodo developed a PHP script that based on iTop's soap web service for creating an incident ticket.

This script is listed in Appendix A and called *createTicket-Nagios.php* it requires the companion file *itopsoatypes.class.inc.php* also listed in Appendix B.

This script is just an example describing how to use iTop soap web service. It can be improved depending on your requirements.

This script is using the hostname of the host that triggered the alarm in Nagios in order to identify the impacted object in iTop. Therefore the hostname has to be unique.

The script triggers tickets only for Nagios alarms that are in HARD state.

Moreover, it requires a default Workgroup to assign the ticket, a default caller and a default customer to be already configured in iTop. (Look at the iTop documentation [here](#) to have more details on Workgroups, Callers and Customers)

The severity of the ticket is always the same, set to critical by default.

All parameters used for ticket creation are listed at the beginning of the script and can be changed depending on your needs.

**Step1:** Copy this script in <yourDirectory> as well as itopsoatypes.class.inc.php

**Step 2:** Define a new Nagios Command:

add following to your Nagios command file (most of the time it is called commands.cfg)

```
# Create incident tickets in iTop command definition
define command{
    command_name    create-iTop-ticket
    command_line    <php path> <yourDirectory>/createTicket-Nagios.php
"$HOSTNAME$" "$SERVICEDESC$" "$SERVICESTATE$" "$SERVICESTATETYPE$"
"$LONGSERVICEOUTPUT$"
}
```

where <php path> is the full path for PHP and <itop path> the path where you installed iTop.

You can test ticket creation by running manually this PHP script.

**Step 3 :** Use this command using an event handler option for each host or service template that should trigger a ticket creation:

You can define it globally for all hosts and services using following options defined most of the time in nagios.cfg:

```
global_host_event_handler=create-iTop-ticket
global_service_event_handler= create-iTop-ticket
```

Or for each host and services using following options:

```
event_handler    create-iTop-ticket
```

event\_handler\_enabled 1

if you choose this latest option, you will have to configure it for each host and service templates you create.

Once done, next time you will have an HARD alarm in Nagios it will create a ticket automatically in iTop.

## How to integrate Nagios display into iTop for a given type of object with iTop 0.9

When you are looking at details for a given object in iTop you can view different tab in the bottom of the frame. You can create a new tab displaying a Nagios frame for the selected object.

For this you have to go in directory *<iTop path>/business/templates*. You have a list of html templates used for different types of object. For instance server.html for the servers.

It looks like :

```
<div class="page_header">
    <itopblock blockclass="MenuBlock" type="popup" encoding="text/oql" label="Actions">SELECT
$class$ WHERE id = $pkey$</itopblock>
    <h1>$class_name$: <span class="hilite">$name$</span></h1>
    <itopblock blockclass="HistoryBlock" type="toggle" encoding="text/oql">SELECT CMDBChangeOp
WHERE objkey = $pkey$ AND objclass = '$class$'</itopblock>
</div>

<itopblock blockclass="DisplayBlock" asynchronous="false" type="bare_details"
encoding="text/oql">SELECT bizServer WHERE id = $pkey$</itopblock>
<itoptabs>
    <itoptab name="Software">
        <itopblock blockclass="DisplayBlock" type="list" encoding="text/oql"
object_id="$pkey$" target_attr="server_id">SELECT bizSoftware WHERE server_id = $pkey$</itopblock>
        </itoptab>
    <itoptab name="Database">
        <itopblock blockclass="DisplayBlock" type="list" encoding="text/oql"
object_id="$pkey$" target_attr="server_id">SELECT bizDatabase WHERE server_id = $pkey$</itopblock>
        </itoptab>
    <itoptab name="Contacts">
        <itopblock blockclass="DisplayBlock" type="links" link_attr="server_id"
object_id="$pkey$" target_attr="contact_id" encoding="text/oql">SELECT lnkContactServer WHERE
server_id = $pkey$</itopblock>
        </itoptab>
</itoptabs>
```

The frame correspondings to tabs is define between `<itoptabs>` and `</itoptabs>`

For each tab you have a block <itoptab> finished by </itoptab>

To add a new tab for Nagios just had in the selected template:

```
<itoptab name="Nagios">
```

```
<iframe width="100%" height="400" src="http://<nagios_path>/cgi-bin/status.cgi?host=$name$">
```

```
</iframe>
```

```
</itoptab>
```

Where <nagios\_path> is the path to your nagios server for instance myserver/nagios.

To have it working you need to make sure that the name of the object you are using in iTop is equal to the hostname of the object in Nagios.

Once done you just have to reload your browser.

The result should be similar to following picture:

The screenshot shows the iTop web interface in Mozilla Firefox. The browser title is "iTop - localhost - Server details - Mozilla Firefox". The page is titled "iTop - localhost - Server details" and shows the user is logged in as 'admin' (Administrator). The main content area displays "Service Status Details For Host 'localhost'".

At the top, there are summary statistics for the host's status:

Up	Down	Unreachable	Pending	Ok	Warning	Unknown	Critical	Pending
0	0	0	0	6	0	0	0	0

Below this, there are buttons for "All Problems" and "All Types" with counts: 0 problems, 1 type.

The main table shows the following service status details:

Host	Service	Status	Last Check	Duration	Attempt	Status Information
localhost	Current Load	OK	06-04-2010 16:05:50	0d 10h 14m 27s	1/4	OK - load average: 0.02, 0.05, 0.01
localhost	Current Users	OK	06-04-2010 16:04:14	42d 13h 29m 10s	1/4	USERS OK - 1 users currently logged in
localhost	HTTP	OK	06-04-2010 16:04:14	0d 1h 13m 20s	1/4	HTTP OK - HTTP/1.1 302 Found - 0.002 second response time
localhost	PING	OK	06-04-2010 16:04:14	5d 4h 0m 32s	1/4	PING OK - Packet loss = 0%, RTA = 0.06 ms
localhost	Root Partition	OK	06-04-2010 16:05:17	42d 13h 27m 18s	1/4	DISK OK - free space: / 1942 MB (21% inode=63%):
localhost	SSH	OK	06-04-2010 16:04:14	5d 3h 58m 18s	1/4	SSH OK - OpenSSH 5.1p1 Debian-Gubuntu2 (protocol 2.0)
localhost	Swap Usage	OK	06-04-2010 16:04:14	42d 13h 26m 3s	1/4	SWAP OK - 100% free (486 MB out of 486 MB)
localhost	Total Processes	OK	06-04-2010 16:04:14	42d 13h 25m 25s	1/4	PROCS OK: 74 processes with STATE = RSZDT

## How to integrate Nagios display into iTop for a given type of object with iTop 1.0

When you are looking at details for a given object in iTop you can view different tab in the top of the frame. You can create a new tab displaying a Nagios frame for the selected object.

For this you have to go in directory <iTop path>/modules/itop-config-mgmt-1.0.o/ and edit file model.itop-config-mgmt.php.

Look for the section class InfrastructureCI and add the following function within this class. This will add a Tab Nagios for all InfrastructureCI on production, when you are not in edit mode.

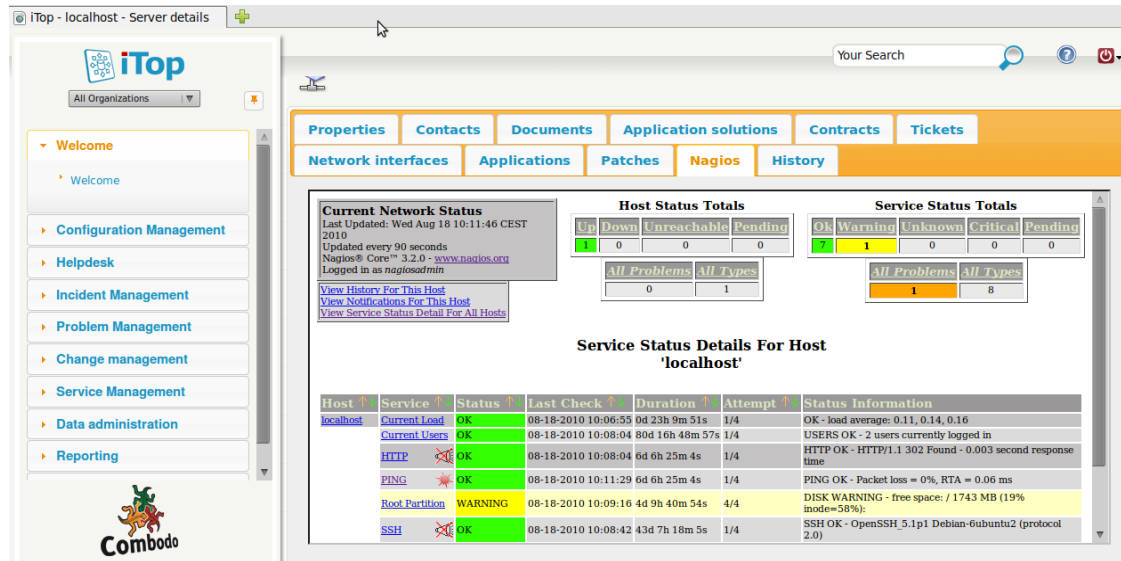
```
function DisplayBareRelations(WebPage $oPage, $bEditMode = false)
{
    parent::DisplayBareRelations($oPage, $bEditMode);
    if (!$bEditMode)
    {
        $oStatus = $this->Get('status');
        $oName = $this->Get('name');
        if ($oStatus == 'production')
        {
            $oPage->SetCurrentTab(Dict::S('Nagios'));
            $oPage->add("<iframe width=\"100%\"
height=\"400\" src=\"http://<nagios_path>/cgi-bin/status.cgi?
host=$oName\"></iframe>");
        }
    }
}
```

Where **<nagios\_path>** is the path to your nagios server for instance myserver/nagios.

To have it working you need to make sure that the name of the object you are using in iTop is equal to the hostname of the object in Nagios.

Once done you just have to reload the page in your browser.

The result should be similar to following picture:



## Appendix A : createTicket-Nagios.php

```
<?php
require_once('../webservices/itopsoaptypes.class.inc.php');

# Parameters from Nagios when executing automatic actions on alarms
$host=$argv[1]; # $HOSTNAME$
$service=$argv[2]; # $$SERVICEDESC$
$serviceStatus=$argv[3]; # $$SERVICESTATE$
$serviceStateType=$argv[4]; # $$SERVICESTATETYPE$
$serviceMessage=$argv[5]; # $LONGSERVICEOUTPUT$

$soapClient = new SoapClient(

### replace URL below by yours !!
"http://localhost/itop-1.0/web/webservices/itop.wsdl.php",
array(
    'trace' => 1,
    'classmap' => $aSOAPMapping, // defined in
itopsoaptypes.class.inc.php
)
);

$Description='The service '.$service.' is in state '.$serviceStatus;
```

```

$initial=$serviceMessage;
echo "$host -$service- -$serviceStatus- -$serviceStateType- -
$serviceMessage-\n";

if ( $serviceStatus != 'OK')
{
switch($serviceStateType)
{
case 'HARD' :

        $oRes = $oSoapClient->CreateIncidentTicket
        (
            'admin', /* login */
            'admin', /* password */
            $Description, /* Title */
            $initial, /* Description */
            new SOAPEExternalKeySearch(array(new
SOAPSearchCondition('name','Dali'))), /* caller */
            new SOAPEExternalKeySearch(array(new
SOAPSearchCondition('name','Demo'))), /* customer */
            new SOAPEExternalKeySearch(array(new SOAPSearchCondition('name', 'Hw
Management'))), /* Service */
            new SOAPEExternalKeySearch(array(new SOAPSearchCondition('name',
'Troubleshooting'))), /* Sub category */
            '', /* product */
            new SOAPEExternalKeySearch(array(new SOAPSearchCondition('name',
'Hardware support'))), /* workgroup */
            array(
                new SOAPLinkCreationSpec(
                    'Device',
                    array(new SOAPSearchCondition('name', $host)),
                    array()
                ),
            ), /* impacted cis */
            '2', /* impact*/
            '2' /* urgency */
        );
echo "<pre>\n";
print_r($oRes);
echo "</pre>\n";

```

```

        print "<pre>\n";
        print "Request: \n".htmlspecialchars($oSoapClient-
>__getLastRequest()) ."\n";
        print "Response: \n".htmlspecialchars($oSoapClient-
>__getLastResponse()) ."\n";
        print "</pre>";
    }
}
?>

```

## Appendix B : itopsoaptypes.class.inc.php

```
<?php
```

```

// Note: the attributes must have the same names (case sensitive) as
// in the WSDL specification

```

```

class SOAPSearchCondition
{
    public $attcode; // string
    public $value; // mixed

    public function __construct($sAttCode, $value)
    {
        $this->attcode = $sAttCode;
        $this->value = $value;
    }
}

class SOAPExternalKeySearch
{
    public $conditions; // array of SOAPSearchCondition

    public function __construct($aConditions)
    {
        $this->conditions = $aConditions;
    }
}

class SOAPAttributeValue
{

```



```

public $attcode; // string
public $value; // mixed

public function __construct($sAttCode, $value)
{
    $this->attcode = $sAttCode;
    $this->value = $value;
}
}
class SOAPLinkCreationSpec
{
    public $class;
    public $conditions; // array of SOAPSearchCondition
    public $attributes; // array of SOAPAttributeValue

    public function __construct($sClass, $aConditions, $aAttributes)
    {
        $this->class = $sClass;
        $this->conditions = $aConditions;
        $this->attributes = $aAttributes;
    }
}
class SOAPLogMessage
{
    public $text; // string

    public function __construct($sText)
    {
        $this->text = $sText;
    }
}
class SOAPResultLog
{
    public $messages; // array of SOAPLogMessage

    public function __construct($aMessages)
    {
        $this->messages = $aMessages;
    }
}

```

```

}

class SOAPLinkCreationSpec
{
    public $class;
    public $conditions; // array of SOAPSearchCondition
    public $attributes; // array of SOAPAttributeValue

    public function __construct($sClass, $aConditions, $aAttributes)
    {
        $this->class = $sClass;
        $this->conditions = $aConditions;
        $this->attributes = $aAttributes;
    }
}

class SOAPLogMessage
{
    public $text; // string

    public function __construct($sText)
    {
        $this->text = $sText;
    }
}

class SOAPResultLog
{
    public $messages; // array of SOAPLogMessage

    public function __construct($aMessages)
    {
        $this->messages = $aMessages;
    }
}

class SOAPLinkCreationSpec
{
    public $class;
    public $conditions; // array of SOAPSearchCondition
    public $attributes; // array of SOAPAttributeValue

```

```

public function __construct($sClass, $aConditions, $aAttributes)
{
    $this->class = $sClass;
    $this->conditions = $aConditions;
    $this->attributes = $aAttributes;
}
}
class SOAPLogMessage
{
    public $text; // string

    public function __construct($sText)
    {
        $this->text = $sText;
    }
} class SOAPResult
{
    public $status; // boolean
    public $result; // array of SOAPResultMessage
    public $errors; // array of SOAPResultLog
    public $warnings; // array of SOAPResultLog
    public $infos; // array of SOAPResultLog

    public function __construct($bStatus, $aResult, $aErrors, $aWarnings,
$aInfos)
    {
        $this->status = $bStatus;
        $this->result = $aResult;
        $this->errors = $aErrors;
        $this->warnings = $aWarnings;
        $this->infos = $aInfos;
    }
}
$aSOAPMapping = array(
    'SearchCondition' => 'SOAPSearchCondition',
    'ExternalKeySearch' => 'SOAPEExternalKeySearch',
    'AttributeValue' => 'SOAPAttributeValue',
    'LinkCreationSpec' => 'SOAPLinkCreationSpec',
    'LogMessage' => 'SOAPLogMessage',

```

```
'ResultLog' => 'SOAPResultLog',
'ResultData' => 'SOAPResultData',
'ResultMessage' => 'SOAPResultMessage',
'Result' => 'SOAPResult',
);
class SOAPResultLog
{
    public $messages; // array of SOAPLogMessage

    public function __construct($aMessages)
    {
        $this->messages = $aMessages;
    }
}
?>
```