

# SAP Webdispatcher on HASI

prepared for

Novell

Disclaimer      Novell, Inc. makes no representations or warranties with respect to the contents or use of this document, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose.

Trademarks      Novell is a registered trademark of Novell, Inc. in the United States and other countries.

                    \* All third-party trademarks are property of their respective owner.

Copyright 2012 Novell, Inc. All rights reserved. No part of this publication may be reproduced, photocopied, stored on a retrieval system, or transmitted without the express written consent of Novell, Inc.

Novell, Inc.	Novell GmbH
404 Wyman Street	Nördlicher Zubringer 9-11
Waltham	40470 Düsseldorf
Massachusetts 02451	
USA	

Prepared By      Fabian Herschel

SAP Webdispatcher on HASI—Implementation Notes  
May, 2009  
Statement of Work: SOW #  
Novell Number: Client #  
Consultants: Fabian Herschel

# Revision History

Version	Date	Editor	Revisions
0.10	05/13/09	Fabian Herschel	First Draft
0.30	05/15/2009	Fabian Herschel	Details of Implementation
0.40	18.05.2009	Fabian Herschel	Correction and Formating

# Reviewer Record

Version	Date	Reviewers Name	Reviewers Role (Peer/ Team Leader / Project Manager)

# Final Approval

Version	Date	Reviewers Name	Reviewers Role (Peer/ Team Leader / Project Manager)

# Contents

<b>1 Executive Summary.....</b>	<b>1</b>
<b>2 Concept and Overview.....</b>	<b>2</b>
<b>3 Implementation.....</b>	<b>4</b>
3.1 SAP Webdispatcher Installation.....	4
3.2 SAP Webdispatcher Configuration.....	5
3.2.1 Start Profile START_W69_wdisphost.....	5
3.2.2 Instance Profile HB2_W69_wdisphost.....	6
3.3 Heartbeat2 Cluster Configuration.....	7
<b>4 Appendix.....</b>	<b>9</b>
4.1 Novell Links.....	9
4.2 SAP Links.....	9
4.3 List of SAP Notes.....	9

## Tables

## Figures

<a href="#"><u>Illustration 1: Cluster resources for the standalone SAP Webdispatcher.....</u></a>	<a href="#"><u>2</u></a>
<a href="#"><u>Illustration 2: Cluster resources for the integrated SAP Webdispatcher.....</u></a>	<a href="#"><u>3</u></a>
<a href="#"><u>Illustration 3: SAP Webdispatcher Resource Group.....</u></a>	<a href="#"><u>7</u></a>

# 1 Executive Summary

This document describes the concept and implementation to integrate the SAP Webdispatcher into a heartbeat2 cluster based on SUSE Linux Enterprise Server 10 with Service Pack 2 and newest updates.

The proof of concept uses version 7.0 SP 13 of the SAP Webdispatcher.

Chapter 2 gives a short overview about the architecture.

Chapter 3 includes a documentation of the implementation itself.

The appendix provides some links to additional Novell and SAP documentations and notes.

## 2 Concept and Overview

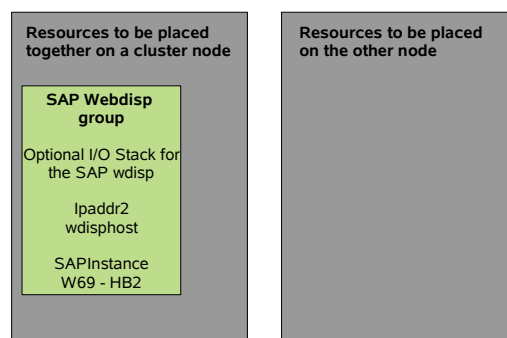
The SAP Webdispatcher is a software to implement load balancing of web applications provided by SAP systems and SAP instances. Clients can contact the SAP Webdispatcher either using the HTTP or HTTPS protocol.

The SAP Webdispatcher provides the following communication modes:

- No encryption - uses HTTP between browser and the SAP Webdispatcher and between the SAP Webdispatcher and the web application server.
- Backend encryption - uses HTTP between browser and the SAP Webdispatcher and HTTPS between the SAP Webdispatcher and the web application server.
- Frontend encryption - uses HTTPS between browser and the SAP Webdispatcher and HTTP between the SAP Webdispatcher and the web application server.
- Complete encryption - uses HTTPS between browser and the SAP Webdispatcher and between the SAP Webdispatcher and the web application server.
- End-To-End encryption - uses HTTPS for the complete communication path, the SAP Webdispatcher only redirects the client requests and server answers.

The first method to use SAP Webdispatcher is to place this component in a separate heartbeat2 cluster in a DMZ or your intranet (short name for this use case is “standalone SAP Webdispatcher”).

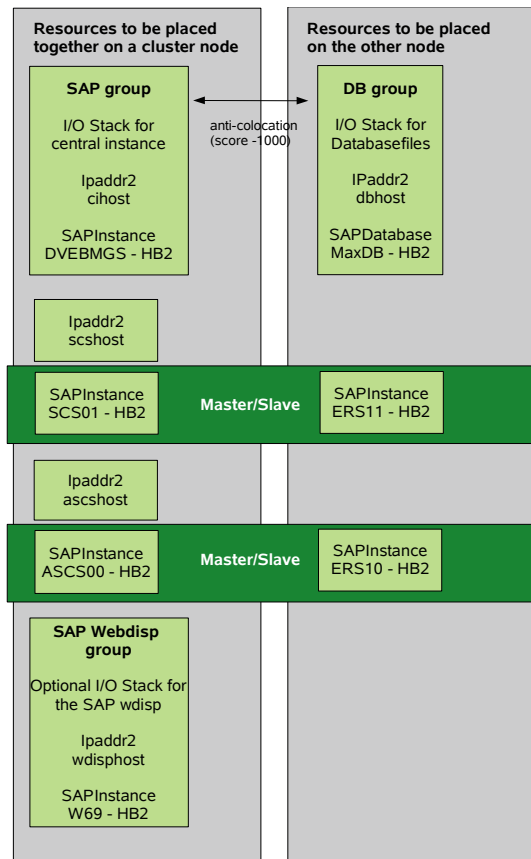
The illustration Cluster resources for the standalone SAP Webdispatcher shows a simplified view of all needed cluster resources for this use case.



**Illustration 1: Cluster resources for the standalone SAP Webdispatcher**

The second method is to integrate the SAP Webdispatcher in an already existing heartbeat cluster running a special SAP system (short name for this use case is “integrated SAP Webdispatcher”).

The illustration shows the enhancement of the use case enqueue-replication of the technical guide SAP Applications Made High Available on SUSE Linux Enterprise Server 10.



**Illustration 2: Cluster resources for the integrated SAP Webdispatcher**

The SAP Webdispatcher is controlled by the heartbeat2 resource agent SAPInstance. This solution might also allow to run multiple instances of the SAP Webdispatcher in a cluster, if you use different ports, users and installation paths for each instance. Please check SAP notes for supported SAP Webdispatcher scenarios.

## 3 Implementation

This chapter describes the installation and configuration of the SAP Webdispatcher in a simplified way. The second aspect is to document the cluster integration. As mentioned before the enqueue replication setup is taken as base, so this chapter will focus on the “integrated SAP Webdispatcher” use case. The most information could also be used for the “standalone SAP Webdispatcher” use case.

### 3.1 SAP Webdispatcher Installation

The SAP Webdispatcher is a component of the SAP kernel and can be downloaded as part of the SAP kernel from the SAP service marketplace. To install the SAP Webdispatcher matching your already installed SAP system, you could also copy the needed files from a SAP instance node.

The following steps should be done on all cluster nodes. If your installation will use the I/O stack (for example the logical volumes and the files system) controlled the cluster you only need to setup the SAP Webdispatcher onetime and create the user accounts on both nodes.

**Disclaimer: These instruction notes describe the installation of the PoC. Please check the SAP installation, user and administration instructions available on <http://service.sap.com> to get a complete installation procedure and to configure the SAP Webdispatcher for productive systems. See SAP note 538405 for more details how to access the installation instructions for several SAP Webdispatcher versions.**

- First of all create a directory `/usr/sap/<SID>/<INSTANCE-NAME>/exe`  
For example: `/usr/sap/HB2/W69/exe`
- Copy the following files into this directory
  - `icmbnd`
  - `libicudata.so.30`
  - `libicui18n.so.30`
  - `libicuuc.so.30`
  - `libsapcrypto.so.30`
  - `libsapcsa.so`
  - `libsapul6_mt.so`
  - `libsapul6.so`
  - `SAPCAR`
  - `sapcpp46.so`
  - `sapstartsrv`
  - `sapwebdisp`
  - `wdispadmin.SAR`
  - `wdispmon`
- To run the SAP Webdispatcher with appropriate user rights you should create an user like `<sid>adm`, `wdisp` or `swdadm`. The user creation is only needed for the described standalone SAP Webdispatcher installation in the DMZ. If you have already installed a SAP instance on the node, there should be already a use `<sid>adm`. To use the `<sid>adm` user is recommended for better assignment to the used SAP installation.

1. Change the file permissions of the copied files, so the above user could run or open them:

```
chown -R <sid>adm:sapsys /usr/sap/<SID>/<INSTANCE-NAME>
```



```
chmod -R u=rwx,g=rx /usr/sap/<SID>/<INSTANCE-NAME>
```

For example:

```
chown -R hb2adm:sapsys /usr/sap/HB2/W69
```

```
chmod -R u=rwx,g=rx /usr/sap/HB2/W69
```

- One of the binaries need to run as user root: `icmbnd`. This binary needs system administrator rights, if you want to bind the SAP Webdispatcher against an IP port below 1024 such as the reserved http port 80.  

```
chown root /usr/sap/wdisp/icmbnd
```

```
chmod u+s /usr/sap/wdisp/icmbnd
```
- You need a virtual IP address and IP hostname which will be used to access the SAP Webdispatcher. In our sample the hostname is `wdisphost` and the IP address is `172.30.100.108`. The hostname and virtual IP address should be added to `/etc/hosts` file of each node to allow local name resolution.

## 3.1 SAP Webdispatcher Configuration

To get an easy cluster integration (see next section), the SAP Webdispatcher has been installed and configured to run in the same mechanism like the other distributed instances of the SAP system for example the ASCS. This means the given solution uses `sapstartsrv` to control the SAP Webdispatcher.

### 3.1.1 Start Profile `START_W69_wdisphost`

The start profile is used by the `SAPInstance` resource agent to tell `sapstartsrv` which service has to be controlled.

The start profile ~~does~~ **uses** variables and references (like `$(variable)`) to minimize the needed configuration changes. The following start profile configures a SAP instance W69 in the SAP system HB2 (the name matches the SAP system name used in the technical guide mentioned before).

The key-value pairs which need to be changed and configured to your SAP system environment are printed in bold.

```
# *****
#.*
#.*      Start profile START_W69_wdisphost
#.*
#.* *****

SAPSYSTEMNAME = HB2
SAPSYSTEM     = 69
INSTANCE_NAME = W69
SAPLOCALHOST = wdisphost

#-----
# Special settings for this manually set up instance
#-----
#
# the webdispatcher does not really need a scs instance number,
# so we set this parameter to the own instance id
#
SCSID = $(SAPSYSTEM)
DIR_INSTANCE = /usr/sap/$(SAPSYSTEMNAME)/$(INSTANCE_NAME)
DIR_EXECUTABLE = $(DIR_INSTANCE)/exe
DIR_PROFILE = /sapmnt/$(SAPSYSTEMNAME)/profile
_PF = $(DIR_PROFILE)/$(SAPSYSTEMNAME)_$(INSTANCE_NAME)_$(SAPLOCALHOST)

# DIR_CT_RUN must be included here to allow the very first run of
# sapcpe in the UNICODE case where it needs the libsapul6 which
# won't be found in an empty DIR_LIBRARY (=DIR_EXECUTABLE)
#
SETENV_00 = LD_LIBRARY_PATH=$(DIR_LIBRARY):$(LD_LIBRARY_PATH)
SETENV_01 = SHLIB_PATH=$(DIR_LIBRARY):$(SHLIB_PATH)
```

```

SETENV_02 = LIBPATH=$(DIR_LIBRARY):$(LIBPATH)

#-----
# Start SAP Webdispatcher
#-----
#
# to get a feasible process name in the process list, we create a
# symbolic link and start the binary using the new name
#
_WDbin = wd.sap$(SAPSYSTEMNAME)_$(INSTANCE_NAME)
_WDpar = -auto_restart -shm_attach_mode 6
Execute_01 = local rm -f $_WD
Execute_02 = local ln -s -f $(DIR_EXECUTABLE)/sapwebdisp $_WD
Start_Program_00 = local $_WDbin pf=$_PF $_WDpar
#-----
# EOF
#-----

```

The given SAP start profile will be referenced later in the heartbeat2 configuration of the cluster resource manager (CRM).

### 3.1.2 Instance Profile HB2\_W69\_wdisphost

The instance profile sets some more key-value pairs. The parameters like `wdisp/*` and `icm/*` are described in the SAP documentation of the SAP Webdispatcher.

In the following the values, which normally should be changed for a basic setup of the SAP Webdispatcher are printed in bold.

This documentation does only focus on the parameters `rdisp/mshost`, `ms/http_port` and `icm/server_port_*`.

- The parameters `rdisp/mshost` and `ms/http` do define the message server connection port. This is needed to tell the SAP Webdispatcher where to get informations about the SAP instances running in the SAP system. If you have a dual stack SAP system using ABAP and Java you should connect the SAP Webdispatcher to the ABAP message server (ASCS instance).
- The parameters `icm/server_port_*` are used to configure the ports and behavior of the SAP Webdispatcher for connections coming from the client side. In our example we define a HTTP port 8080 and a secure port 8483.

```

# unique instance number
SAPSYSTEM = 69
SAPSYSTEMNAME = HB2
INSTANCE_NAME = W69

# add default directory settings
DIR_EXECUTABLE = /usr/sap/HB2/W69/exe
DIR_INSTANCE = /usr/sap/HB2/W69

# Accessibility of Message Servers
rdisp/mshost = ascshost
ms/http_port = 8100

# SAP Web Dispatcher Parameter
wdisp/auto_refresh = 120
wdisp/max_servers = 100

wdisp/shm_attach_mode = 6

# configuration for default scenario (medium size)
icm/max_conn = 500
icm/max_sockets = 1024
icm/req_queue_len = 500
icm/min_threads = 10
icm/max_threads = 50
mpi/total_size_MB = 80

#maximum number of concurrent connections to one server

```

```

wdisp/HTTP/max_pooled_con = 500
wdisp/HTTPS/max_pooled_con = 500

# SAP Web Dispatcher Ports
icm/server_port_0 = PROT=HTTP,PORT=8080,EXTBIN=1
icm/server_port_1 = PROT=ROUTER,PORT=8443,EXTBIN=1
icm/server_port_2 = PROT=HTTPS,PORT=8443,EXTBIN=1

# SAP Web Dispatcher Web Administration
#icm/HTTP/admin_0 = PREFIX=/sap/wdisp/admin,DOCROOT=./admin,AUTHFILE=icmauth.txt
icm/HTTP/admin_0 = PREFIX=/usr/sap/wdisp/admin,DOCROOT=./admin,PORT=8888,HOST=ls3197v6

```

## 3.1 Heartbeat2 Cluster Configuration

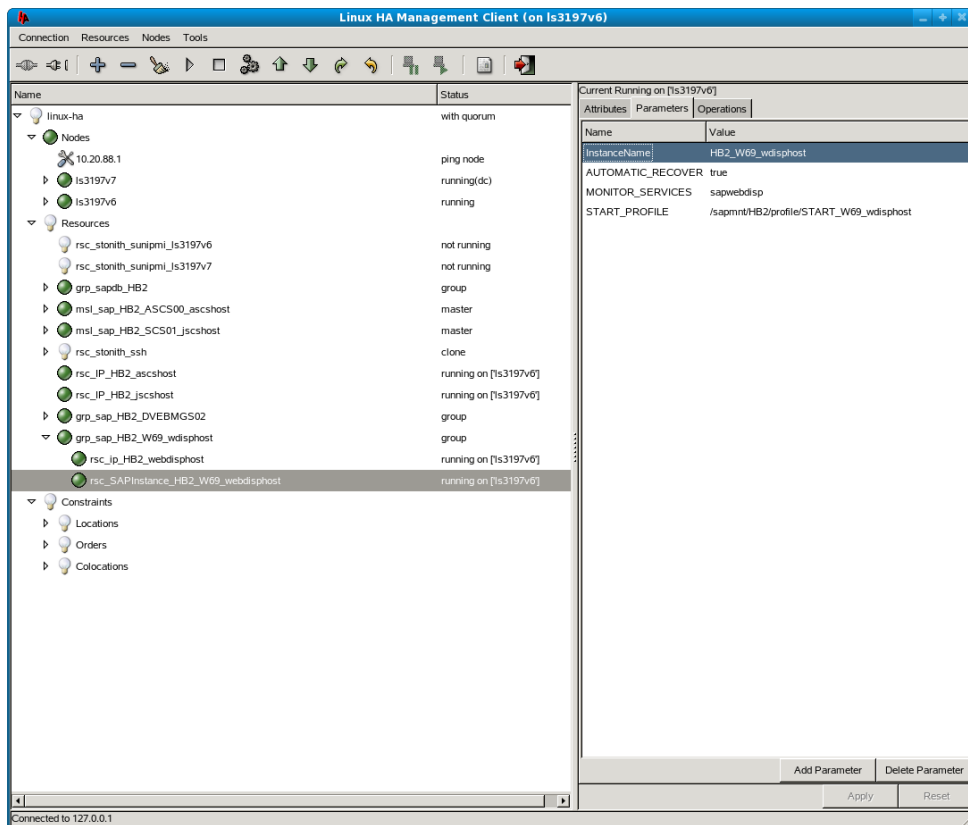
The solution to integrate the SAP Webdispatcher into a heartbeat2 cluster needs at least the version 2.01 of the resource agent SAPInstance<sup>1</sup>. This version is newer than the version included in SLES10SP2. Version 2.01 provides the following new features (in comparison with version 1.92):

- The resource agent provides the actions promote and demote for master/slave resources (such as included in the enqueue replication use case).
- The resource agent allows to configure the services to be monitored and controlled by the resource agent.

To integrate the SAP Webdispatcher in a new or already running heartbeat2 cluster, you only need to add a new resource group controlling all needed resources such as a virtual IP address, optional file systems and the SAP Webdispatcher instance itself. This sample only places the resources IP and SAPInstance in the SAP Webdispatcher resource group. For productive environments it is recommended to use a shared file system as described in the technical guide SAP Applications Made High Available on SUSE Linux Enterprise Server 10.

This chapter provides a sample XML-part which defines such a resource group for our sample setup. The parts, which have to be adapted to your cluster environment are printed in bold.

<sup>1</sup> The newest versions of SAPInstance and SAPDatabase will be available for download at SourceForge.net (search for sapagents). Supported versions are available via the normal Novell update services. To cover the gap between the newest supported version available via update services and needed features ask NTS for the support status.



**Illustration 3: SAP Webdispatcher Resource Group**

You should at least set:

- The group id (this name will also be viable in the cluster gui hb\_gui)
- The virtual IP address of the service
- The SAP instance name (here HB2\_W69\_wdisphost)
- The list of services to be monitored (the value sapwebdisp should fit normal needs)
- The start profile (this is parameter is optional, but it could be helpful to define the path explicitly)

Additional to the basic setup you might need to tune the operation timeouts.

```
<group id="grp_sap_HB2_W69_wdisphost">
  <meta_attributes id="grp_sap_HB2_W69_wdisphost_meta_attrs">
    <attributes>
      <nvpair id="grp_sap_HB2_W69_wdisphost_metaattr_ordered" name="ordered"
value="true"/>
      <nvpair id="grp_sap_HB2_W69_wdisphost_metaattr_collocated" name="collocated"
value="true"/>
    </attributes>
  </meta_attributes>
  <primitive id="rsc_ip_HB2_webdisphost" class="ocf" type="IPAddr2"
provider="heartbeat">
    <instance_attributes id="rsc_ip_HB2_webdisphost_instance_attrs">
      <attributes>
        <nvpair id="rsc_ip_HB2_webdisphost_attr_0" name="ip" value="172.30.100.108"/>
      </attributes>
    </instance_attributes>
  </primitive>
  <primitive class="ocf" type="SAPInstance" provider="heartbeat"
id="rsc_SAPInstance_HB2_W69_webdisphost">
    <instance_attributes id="rsc_SAPInstance_HB2_W69_webdisphost_instance_attrs">
      <attributes>
        <nvpair id="rsc_SAPInstance_HB2_W69_webdisphost_attr_0" name="InstanceName"
value="HB2_W69_wdisphost"/>
        <nvpair id="rsc_SAPInstance_HB2_W69_webdisphost_attr_1"
```

```

name="AUTOMATIC_RECOVER" value="true"/>
  <nvpair id="rsc_SAPInstance_HB2_W69_webdisphost_attr_2"
name="MONITOR_SERVICES" value="sapwebdisp"/>
  <nvpair id="rsc_SAPInstance_HB2_W69_webdisphost_attr_3" name="START_PROFILE"
value="/sapmnt/HB2/profile/START_W69_wdisphost"/>
  </attributes>
</instance_attributes>
<operations>
  <op id="rsc_SAPInstance_HB2_W69_webdisphost_start" name="start" timeout="60"
start_delay="0" disabled="false" role="Started"/>
  <op id="rsc_SAPInstance_HB2_W69_webdisphost_stop" name="stop" timeout="60"
start_delay="0" disabled="false" role="Started" on_fail="block"/>
  <op name="monitor" interval="10" timeout="30" disabled="false" role="Started"
id="rsc_SAPInstance_HB2_W69_webdisphost_mon" start_delay="5"/>
</operations>
<meta_attributes id="rsc_SAPInstance_HB2_W69_webdisphost_meta_attrs">
  <attributes/>
</meta_attributes>
</primitive>
</group>

```

If you have changed the sample to fit your cluster environment, you can add the resource group using the cib administration command `cibadmin`.

```
cibadmin -C -o resources -x <yourfile.xml>
```

# 1 Appendix

## 1.1 Novell Links

Main page about Novell and SAP:

<http://www.novell.com/sap>

Technical guide “SAP Applications Made High Available on SUSE Linux Enterprise Server 10”:

[http://www.novell.com/partners/sap/sap\\_on\\_hasi.pdf](http://www.novell.com/partners/sap/sap_on_hasi.pdf)

## 1.2 SAP Links

SAP Service Marketplace:

<http://service.sap.com>

SAP Download at Service Marketplace:

<http://service.sap.com/swdc>

SAP Installation guides at Service Marketplace:

<http://service.sap.com/installNW70>

SAP Notes at Service Marketplace:

<http://service.sap.com/notes>

## 1.3 List of SAP Notes

<b>538405</b>	<a href="#">Composite SAP Note: SAP Web Dispatcher</a>
<b>974284</b>	<a href="#">SAP Web Dispatcher 7.00: Patch history</a>
<b>908097</b>	<a href="#">SAP Web Dispatcher 7.00: Importing patches</a>

< Do Not Delete the Line Feeds after the next Paragraph>

<End of Document>

