

Novell[®] Sentinel[™]

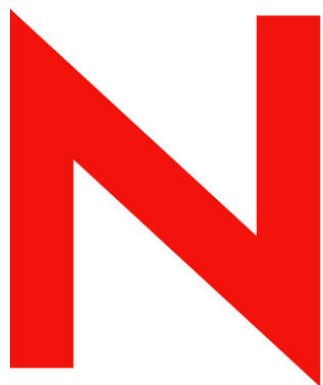
6.0

September 7, 2007

www.novell.com

Using Sentinel 5.x Collectors in Sentinel 6.0

Product Version(s): Requires Sentinel 6.0 or higher



Novell[®]

Legal Notices

Novell, Inc. makes no representations or warranties with respect to the contents or use of this documentation, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. Further, Novell, Inc. reserves the right to any and all parts of Novell software, to revise this publication and to make changes to its content, at any time, without obligation to notify any person or entity of such revisions or changes.

Any products or technical information provided under this Agreement may be subject to U.S. export controls and the trade laws of other countries. You agree to comply with all export control regulations and to obtain any required licenses or classification to export, re-export, or import deliverables. You agree not to export or re-export to entities on the current U.S. export exclusion lists or to any embargoed or terrorist countries as specified in the U.S. export laws. You agree to not use deliverables for prohibited nuclear, missile, or chemical biological weaponry end uses. Please refer to <http://www.novell.com/info/exports/> for more information on exporting Novell software. Novell assumes no responsibility for your failure to obtain any necessary export approvals.

Copyright © 1999-2007 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 the publisher.

Novell, Inc. has intellectual property rights relating to technology embodied in the product that is described in this document. In particular, and without limitation, these intellectual property rights may include one or more of the U.S. patents listed at <http://www.novell.com/company/legal/patents/> and one or more additional patents or pending patent applications in the U.S. and in other countries.

Novell, Inc.
404 Wyman Street, Suite 500
Waltham, MA 02451
U.S.A.
<http://www.novell.com>

Online Documentation: To access the online documentation for this and other Novell products and to get updates, see <http://www.novell.com/documentation>.

Novell Trademarks

For Novell trademarks, see the Novell Trademark and Service Mark list (<http://www.novell.com/company/legal/trademarks/tmlist.html>).

Third-Party Materials

All third-party trademarks are the property of their respective owners.

Third Party Legal Notices

This product may include the following open source programs that are available under the LGPL license. The text for this license can be found in the Licenses directory.

- edtFTPj-1.2.3 is licensed under the Lesser GNU Public License. For more information, disclaimers and restrictions see <http://www.enterprisedt.com/products/edftpj/purchase.html>.
- Esper. Copyright © 2005-2006, Codehaus.
- jTDS-1.2.jar is licensed under the Lesser GNU Public License. For more information, disclaimers and restrictions see <http://jtds.sourceforge.net/>.
- MDateSelector. Copyright © 2005, Martin Newstead, licensed under the Lesser General Public License. For more information, disclaimers and restrictions see <http://web.ukonline.co.uk/mseries>.
- Enhydra Shark, licensed under the Lesser General Public License available at: <http://shark.objectweb.org/license.html>.
- Tagish Java Authentication and Authorization Service Modules, licensed under the Lesser General Public License. For more information, disclaimers and restrictions see <http://free.tagish.net/jaas/index.jsp>.

This product may include software developed by The Apache Software Foundation (<http://www.apache.org/>) and licensed under the Apache License, Version 2.0 (the "License"); the text for this license can be found in the Licenses directory or at <http://www.apache.org/licenses/LICENSE-2.0>. Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

The applicable open source programs are listed below.

- Apache Axis and Apache Tomcat, Copyright © 1999 to 2005, Apache Software Foundation. For more information, disclaimers and restrictions, see <http://www.apache.org/licenses/>.
- Apache Lucene, Copyright © 1999 to 2005, Apache Software Foundation. For more information, disclaimers and restrictions, see <http://www.apache.org/licenses/>.
- Bean Scripting Framework (BSF), licensed by the Apache Software Foundation Copyright © 1999-2004. For more information, disclaimers and restrictions see <http://xml.apache.org/dist/LICENSE.txt>.
- Skin Look and Feel (SkinLF). Copyright © 2000-2006 L2FProd.com. Licensed under the Apache Software License. For more information, disclaimers and restrictions see <https://skinlf.dev.java.net/>.
- Xalan and Xerces, both of which are licensed by the Apache Software Foundation Copyright © 1999-2004. For more information, disclaimers and restrictions see <http://xml.apache.org/dist/LICENSE.txt>.

This product may include the following open source programs that are available under the Java license.

- JavaBeans Activation Framework (JAF). Copyright © Sun Microsystems, Inc. For more information, disclaimers and restrictions see <http://www.java.sun.com/products/javabeans/glasgow/jaf.html> and click download > license.
- Java 2 Platform, Standard Edition. Copyright © Sun Microsystems, Inc. For more information, disclaimers and restrictions see <http://java.sun.com/j2se/1.5.0/docs/relnotes/SMICopyright.html>.
- JavaMail. Copyright © Sun Microsystems, Inc. For more information, disclaimers and restrictions see <http://www.java.sun.com/products/javamail/downloads/index.html> and click download > license.

This product may also include the following open source programs.

- ANTLR. For more information, disclaimers and restrictions, see <http://www.antlr.org>.
- Boost. Copyright © 1999, Boost.org.
- Concurrent, utility package. Copyright © Doug Lea. Used without CopyOnWriteArrayList and ConcurrentReaderHashMap classes.
- Java Ace, by Douglas C. Schmidt and his research group at Washington University. Copyright © 1993-2005. For more information, disclaimers and restrictions see <http://www.cs.wustl.edu/~schmidt/ACE-copying.html> and <http://www.cs.wustl.edu/~pjain/java/ace/JACE-copying.html>.
- Java Service Wrapper. Portions copyrighted as follows: Copyright © 1999, 2004 Tanuki Software and Copyright © 2001 Silver Egg Technology. For more information, disclaimers and restrictions, see <http://wrapper.tanukisoft.com/doc/english/license.html>.
- JLDAP. Copyright 1998-2005 The OpenLDAP Foundation. All rights reserved. Portions Copyright © 1999 - 2003 Novell, Inc. All Rights Reserved.
- OpenSSL, by the OpenSSL Project. Copyright © 1998-2004. For more information, disclaimers and restrictions, see <http://www.openssl.org>.
- Rhino. Usage is subject to Mozilla Public License 1.1. For more information, see <http://www.mozilla.org/rhino/>.
- Tao (with ACE wrappers) by Douglas C. Schmidt and his research group at Washington University, University of California, Irvine and Vanderbilt University. Copyright © 1993-2005. For more information, disclaimers and restrictions see <http://www.cs.wustl.edu/~schmidt/ACE-copying.html> and <http://www.cs.wustl.edu/~pjain/java/ace/JACE-copying.html>.
- Tinyxml. For more information, disclaimers and restrictions see <http://grinninglizard.com/tinyxmldocs/index.html>.

NOTE: As of the publication of this documentation, the above links were active. In the event you find that any of the above links are broken or the linked web pages are inactive, please contact Novell, Inc., 404 Wyman Street, Suite 500, Waltham, MA 02451 U.S.A.

Preface

This manual gives you a general understanding of this Connector and the differences between this connection method in Sentinel 6 and previous versions of Sentinel. It is intended mainly for the system administrators to configure the Connector, to establish connection between Collectors and Event Source.

Additional Stopgap documentation available on Novell Web Portal are:

- Sentinel 6.0 Syslog Connector Guide
- Sentinel 6.0 Audit Connector Guide
- Sentinel 6.0 DB Connector Guide
- Sentinel 6.0 File Connector Guide
- Sentinel 6.0 WMI Connector Guide
- Using 5.x Collectors in Sentinel 6.0

Feedback

We want to hear your comments and suggestions about this manual and the other documentation included with this product. Please use the User Comments feature at the bottom of each page of the online documentation and enter your comments there.

Additional Documentation

The other manuals on this product are available at <http://www.novell.com/documentation>.

For additional documentation to install and use Connectors and Collectors, see [Sentinel User Guide](#).

Documentation Conventions

Notes and Cautions

NOTE: Notes provide additional information that may be useful.

WARNING:

Warning provides additional information that may keep you away from performing tasks that may cause damage or loss of data.

Commands

Commands appear in courier font. For example:

```
useradd -g dba -d /export/home/oracle -m -s /bin/csh
oracle
```

Internal References:

- For more information, see “[Section Name](#)” (if in the same Chapter).
- For more information, see [Chapter number](#), “[Chapter Name](#)” (if in the same Guide).
- For more information, see [Section Name in Chapter Name](#), [Guide Name](#) (if in a different Guide).

Other References

The following manuals are available with the Sentinel install CDs.

- Sentinel Install Guide
- Sentinel User Guide
- Sentinel Collector Builder User Guide
- Sentinel User Reference Guide
- Sentinel 3rd Party Integration Guide
- Release Notes

Contacting Novell

- Website: <http://www.novell.com>
- Novell Technical Support:
http://support.novell.com/phone.html?sourceidint=suplnav4_phonesup
- Self Support:
http://support.novell.com/support_options.html?sourceidint=suplnav_supportprog
- Patch Download Site: <http://download.novell.com/index.jsp>
- 24x7 support: <http://www.novell.com/company/contact.html>
- For Collectors/Connectors/Reports/Correlation/Hotfixes/TIDS:
<http://support.novell.com/products/sentinel>

Contents

Introduction	1
Differences between Sentinel 5.x and Sentinel 6	1
Differences in Functionality	2
Connection Methods	2
RX State.....	3
s_RXBufferString and i_RXBufferLength	3
Delimiter	3
Timeout	4
Heartbeat	4
Collectors that Exit	4
Collectors with File Rotation.....	4
Paths and Directories	4
Collector Script Commands.....	5
SHELL	5
FILE Commands	5
GETCONFIG	5
Deprecated Commands.....	6
Building Collectors and Parameter Usage.....	6
Importing Collectors	6
Multiple Instances of the same Collector.....	7
A Revision History	A-1
Revision 01.....	A-1

Introduction

Sentinel 6.0 provides a graphical Event Source Management framework which helps in deploying, managing, and troubleshooting Collectors within the Sentinel console. This framework replaces functionality previously in the Sentinel Collector Builder and provides new features.

The following table shows some of the major functions in Sentinel and what application is used to perform those functions.

Function	Sentinel 5.1.3 and previous versions	Sentinel 6.0
Build Collector	Collector Builder	Collector Builder
Modify Collector	Collector Builder	Collector Builder
Debug Collector	Collector Builder	Sentinel Control Center
Deploy Collector	Collector Builder	Sentinel Control Center
Set Parameter Values	Collector Builder	Sentinel Control Center

This guide assumes that you are familiar with:

- Collector and Connectors in Sentinel 5.x
- Importing and configuring Collectors and Connectors in Sentinel 5.x

Some Collectors written for Sentinel 5.x and previous versions of Sentinel (referred as “Sentinel 5.x”) may require modification in order to work with Sentinel 6.0. The purpose of this document is to provide information in order to help remediate potential problem areas and help port custom Collectors to Sentinel 6.0. Some issues to consider include:

- Degree of customization to the Collector and availability of a comparable Sentinel 6 Collector from Novell
- Legacy connection method vs. Connector plug-in
- JDBC vs. ODBC connection
- GETCONFIG command changes
- FILE command changes
- SHELL commands and associated paths
- Legacy heartbeat messages
- Multiple instances of a Collector, implemented using multiple .chn files
- Non-default record delimiter in Collectors using the FILE connection method

This document must be used in conjunction with several other documentation sources listed in [“Additional Documentation”](#).

Differences between Sentinel 5.x and Sentinel 6

Due to changes in functionality between Sentinel 5.x and Sentinel 6, the following are not valid for Sentinel 6 Collectors and Connectors.

- Port configuration
- Installing the DB Connector
- Information about setting parameters (actual parameter names and values are still valid, but the method for setting them has changed).

Differences in Functionality

The several differences in functionality between the the Connector for Sentinel 5.x and Sentinel 6.0 are explained below.

Connection Methods

In Sentinel 5.x, the connectivity code for file, serial, socket, ODBC, SNMP and process connections was embedded in the Collector Engine and Collector Manager processes. Other connectivity methods (for example, WMI, JDBC, SDEE, Syslog and LEA) were handled by creating an external process and connecting to that process using the internal process Connector. Collectors were built, modified, and deployed using the Collector Builder application.

In Sentinel 6.0, parameter value defaults are set in Collector Builder, but the actual values for a deployed Collector are set in the Sentinel Control Center.

In Sentinel 6.0, Connector plugins have replaced the embedded connection methods. This standardizes the way Sentinel acquires data from devices that it monitors and provides the ability to easily perform updates when a newer version of the Connector is released. It is possible to use the Sentinel 6.0 Process Connector plugin to continue using a custom connection method from Sentinel 5.x, but some features of Event Source Management will be unavailable. To take advantage of all of the new features of Event Source Management, the old Sentinel 5.x connection methods should be replaced by Sentinel 6.0 Connector plug-ins.

The following table contains the available Sentinel 5.x Connectors and the equivalent Connector plugins for Sentinel 6.0.

Sentinel 5	Sentinel 6 .0
File Connector internal to Collector Manager	File Connector plugin
Serial Connector internal to Collector Manager	No current support planned
Socket Connector internal to Collector Manager	Socket Client Connector plugin (not yet released)
ODBC script commands internal to Collector Manager	Supported, but does not use either a Connector or Event Source. Only the Collector is needed. However, it is highly recommend that the code be modified to support the DB Connector instead.
Process Connector internal to Collector Manager	Process Connector plugin
SNMP Connector available through port configuration, internal to Collector manager	SNMP Connector plugin
WMI using EventLog.exe and the process Connector	WMI Connector plugin
JDBC using dbconnector.bat or dbconnector.sh and the process Connector	Database (DB) Connector plug-in
SDEE using sdee-app.bat or sdee-app.sh and the process Connector	SDEE Connector plug-in

Syslog Connector using SyslogConnectorAgent.bat or SyslogConnectorAgent.sh and the process Connector	Syslog Connector plug-in
Audit Connector using SyslogConnectorAgent.bat or SyslogConnectorAgent.sh, modified for Audit, and the process Connector	Audit Connector plug-in
LEA Connector using lea_client.exe and the process Connector	LEA Connector plug-in
Custom application using the process Connector	Process Connector plug-in and custom application

For more information about any of these Connectors, see [“Additional Documentation”](#).

RX State

In Sentinel 5.x, the RX State had various functionalities. It allowed the person creating the Collector to specify the delimiter (to indicate the end of a record) and set the timeout limit for receiving data. During operation, the RX State internally set the values of two variables when it returned control to the parsing script. In Sentinel 6.0, some of these functionalities have changed.

s_RXBufferString and i_RXBufferLength

In Sentinel 5.x, the RX State internally set the values of two variables, s_RXBufferString and i_RXBufferLength, when it returned control to the parsing script.

In Sentinel 6.0, the same variables are set, using the same format, for backward compatibility. For example, the DB Connector plug-in still returns name-value pair data in s_RXBufferString. In addition, the Connector plugging may also set additional variables, which are described in Connector specific documentation.

The addition of new variables to the Connector plugging should not have a negative impact on Sentinel 5.x Collectors running in Sentinel 6.0. However, it is a good practice to check the Connector documentation for the names of these new variables and ensure that your Collector script does not use the same variable names.

Delimiter

In Sentinel 5.x, the RX state specified a record delimiter, which was embedded in the parsing script. For deployed Collectors, this information can be viewed in the Sentinel 5.x Collector Builder application.

In Sentinel 6.0, the original delimiter is not available to the Connector plug-in and must be specified in the Connector plugin’s configuration. This may be done at deployment time or by setting connection mode properties in the Collector’s package.xml file. The supported set of properties can be found in the respective Connector plug-in documents.

Timeout

In Sentinel 5.x, the RX state can be set to timeout if it is of type TIMEOUT or DELIMITED TIMEOUT. This information is embedded in the parsing script and can be viewed in the Sentinel 5.x Collector Builder application. The occurrence of a timeout would cause a returning RX state with `i_RXBufferLength` set to 0 and a blank `s_RXBufferString`. Many Sentinel 5.x Collectors performed an action when the timeout was detected they can:

- Send an internal event called a heartbeat
- Stop processing and exit (for example, when a vulnerability scan read is complete)
- Start reading from a new input file (file rotation)

In Sentinel 6.0, the timeout period can be set for older scripts utilizing the timeout capability of the RX state when deploying the Collector by setting the `RX_TIMEOUT_DELAY` parameter.

Heartbeat

In Sentinel 5.x, many Collectors sent a heartbeat when a timeout happened. This indicated that although no data was received, the Collector was still up and running. In Sentinel 6.0, the Event Source Management framework monitors Collectors, Connectors, and Event Sources and displays status information. Therefore, the heartbeat messages in Sentinel 5.x Collectors may be considered redundant. In most Sentinel 5.x Collectors, the heartbeat messages may be disabled in Sentinel 6.0 when the Collector is deployed by setting the parameter `Send_Heartbeat_Messages` to `off`.

Collectors that Exit

In Sentinel 5.x, some Collectors are designed to run and then automatically exit using a Stop state in the Collector template. For example, a vulnerability Collector may be designed to run until it reaches the end of the file (indicated by the timeout), and then stop.

In Sentinel 6.0, the Event Source Management framework does not recognize the Stop state. It may show an error for a Collector that intentionally ran and then exited. You can manually stop the Collector in the *ESM Live View*.

Collectors with File Rotation

In Sentinel 5.x, some Collectors are designed to use the timeout to indicate that the end of the log file had been reached and it was necessary to rotate to a new log file. Because the internal file Connector is no longer used in Sentinel 6.0, using the GETCONFIG command in Sentinel 6.0 for file rotation will not work. For more information about how to use the log file rotation functionality built into the Connector Plug-in, see the [Connector documentation or File Connector Differences in Sentinel 6.0](#).

Paths and Directories

In Sentinel 5.x, a running instance of a Collector shared the same file space as the Collector code:

```
%ESEC_HOME%\wizard\Elements
$ESEC_HOME/wizard/Elements
```

The environment variable WORKBENCH_HOME referred to %ESEC_HOME%\wizard.

In Sentinel 6.0, the running instance and the development instance have been separated into two different locations

```
%ESEC_HOME%\data\collector_workspace (for
development)
%ESEC_HOME%\data\collector_mgr.cache\collector_in
stances (for running instances)
```

The value for the environment WORKBENCH_HOME has been changed from %ESEC_HOME%\wizard (in Sentinel 5.x) to simply %ESEC_HOME% (in Sentinel 6.0).

In Sentinel 6.0, a new directory is created called %ESEC_HOME%\data. This directory has several subdirectories with the same names as the Collectors. The data directory is used to store data that needs to be preserved after the Collector is modified, such as an offset file. (If a Collector is modified and redeployed, the running directory in %ESEC_HOME%\data\collector_mgr.cache\collector_instances will also be modified.)

These path and directory changes may affect Collector script implementations.

Collector Script Commands

Several commands are modified or deprecated in Sentinel 6.0. All Sentinel 5.x Collectors should be examined to determine whether they use these commands.

SHELL

Collectors that use the SHELL command should be examined to ensure the directory path is valid. Any Collector that uses an absolute directory path or a relative path that does not strictly traverse down a path (for example, one that uses ..\..\), should be examined for correctness in the Sentinel 6.0 environment.

FILE Commands

In Sentinel 6.0, the FILEA, FILER, FILEW, and FILEL commands have all been modified to work from the %ESEC_HOME%\data directory.

GETCONFIG

The GETCONFIG command has been updated to use the Sentinel 6.0 paths. The following properties have been updated:

- System.PortScript returns the running instance directory name (for example, WMI_6_0_Collector_68714633-A987-1029-A520-000C29F2D765, which is a combination of the script Collector name and the instance UUID)
- System.Data_Dir returns the path to the data directory (%ESEC_HOME%\data)
- System.Agent_Dir returns the path to the parent directory of all the running Collector instances (for example, %ESEC_HOME%\data\collector_mgr.cache\collector_instances)

- `System.Local.Dir` returns the path to the Collector's running instance directory (equivalent to `System.Agent.Dir` concatenated with `System.PortScript`)

Deprecated Commands

In Sentinel 6 several parsing commands have been deprecated.

- `DISPLAY` – replaced by debugging functionality in Sentinel 6.0
- `INDICATOR` – replaced by the `EVENT` command
- `POPUP` – replaced by debugging functionality in Sentinel 6.0

Although the DB commands (`DBOPEN`, `DBGETROW`, and others) used in ODBC Collectors are still functional in Sentinel 6, Novell recommends that ODBC Collectors be updated to connect through JDBC instead. The advantages of using JDBC instead of ODBC are explained in the *DBConnector Differences Guide*.

Building Collectors and Parameter Usage

In Sentinel 5.x, after modifying a Collector in the Collector Builder, it was necessary to manually “build” the Collector and then upload it to the Collector Manager. During the build:

- The parameter values were saved to a file with a `.par` extension.
- A script file was written to a file with an `.asd` extension.

In Sentinel 6.0 Collectors are automatically built before execution by the Event Source Management framework. The parameter values entered when the Collector is deployed are stored in the database and in the `parameters.csv` file in the Collector's running instance directory before execution. This set of values is built into `.asd` file automatically when the Collector is executed.

When a Sentinel 5.x Collector is imported into Sentinel 6.0, the names of the parameters will be standardized. These name changes will be visible in Collector Builder.

WARNING: The Collector may stop functioning if you change the new parameter names.

Importing Collectors

After completing any necessary Collector modifications, you can import the 5.x Collector into Sentinel 6.0.

To import a Collector from an existing 5.x implementation in Sentinel 6.0:

1. Install Sentinel Control Center (version 6.0) on the machine with the Collector files or copy the Collector files to a machine on which the Sentinel Control Center (version 6.0) is installed.
2. Open Sentinel Control Center as a user with the permission to use Event Source Management.
3. Go to Event Source Management > Live View.
4. Import a Connector. For more information, see *Sentinel User Guide*.
5. Import a Collector from a directory. For more information, see *Sentinel User Guide*.

6. Complete the fields for any missing metadata.

Multiple Instances of the same Collector

In Sentinel 5, there were two ways to support multiple instances of the same Collector code. The most common method was to copy the Collector and modify the parameters of the copied Collector. This method is still supported in Sentinel 6.0.

In Sentinel 5.x, the other way to support multiple instances of the same Collector was to add a new column to the parameter configuration in Collector Builder. This information was saved to the parameters (.par) file. A new .asd script file was created automatically during the build process, and the Collector developer had to manually create a new .chn file that referred to the new .asd file. The new .chn file was then selected during the port configuration when deploying the Collector.

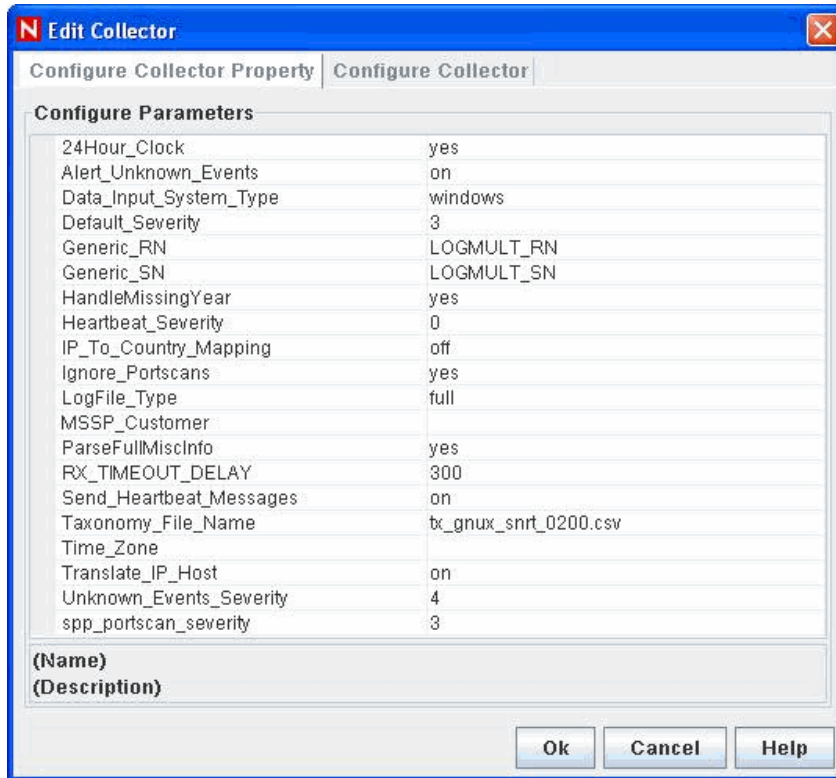
In Sentinel 6.0, this method of deploying two instances of a Collector is not supported. Only the first set of parameter values from the .par file will be recognized when the Collector is imported, and that set will be used to set the default values for the Collector. To achieve the similar functionality in Sentinel 6.0, you must deploy the Collector twice, once with each set of parameter values.

To determine whether you have any Sentinel 5.x Collectors that were using this configuration, you can search the Collector directories with more than one .chn file. on the Collector Manager(s).

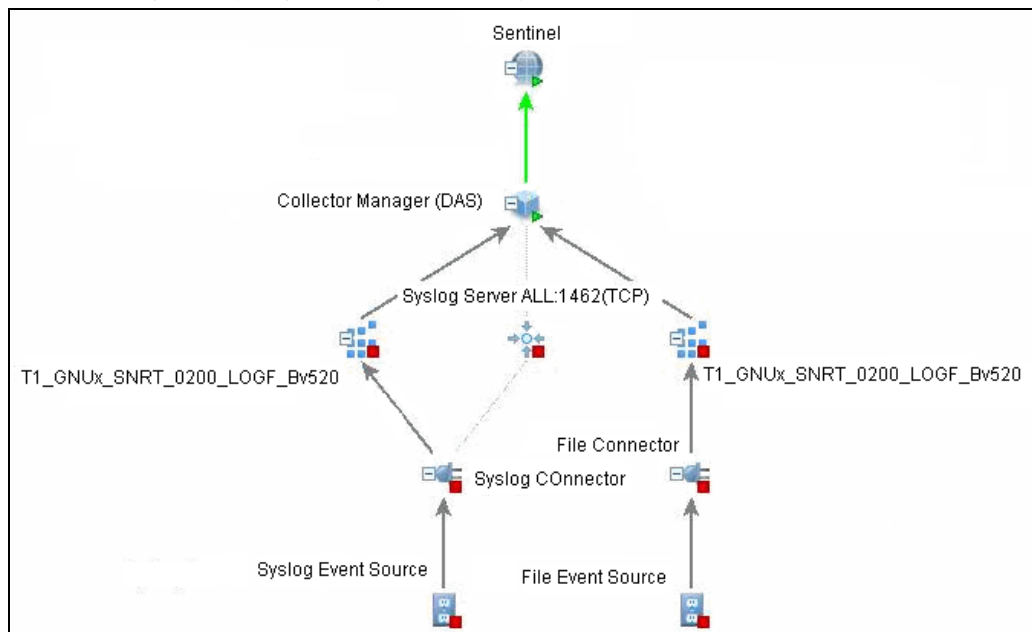
The instructions below assume that you have the knowledge of importing and deploying Collectors and Connectors in Sentinel 6.0 and the usage of the Collector Builder. For more information, see [Event Source Management in Sentinel User Guide](#).

To deploy two instances of a Collector:

1. Log into the Sentinel Control Center as a user with permission to deploy Collectors.
2. Select Event Source Management > Live View.
3. Import the Collector. For more information, see [Sentinel User Guide](#).
4. Import the required Connector. For more information, see [Sentinel User Guide](#).
5. Deploy two instances of the Collector, using the same imported Collector but setting the parameter values appropriately for each instance. The following screen shows one set of parameter values.



- Configure one or more event sources, as required for each of the two Collector instances. The final configuration for two instances of the Sentinel 5.x Snort Collector, one created using the “full” configuration and one using the “syslog” configurations may look like this:



NOTE: The values used in Sentinel 5.x for the second instance of a deployed Collector may be viewed in Collector Builder on the Parameters screen.

APPENDIX

A Revision History

Revision 01

Initial Document

July 2007