



# Using the Signature Editor

---

Published: August 23, 2013, OL-29110-05

## Introduction

This module describes the Signature Editor tool and how to use it to create and modify Dynamic Signature Script (DSS) files.

The Signature Editor tool allows you to create and modify DSS files that can add and modify protocols and protocol signatures in the Cisco SCA BB, based on your knowledge of new network protocols that Cisco SCA BB is yet to support.

This chapter consists of these sections:

- [The Signature Editor Console, page 13-1](#)
- [Managing DSS Files, page 13-2](#)
- [How to Create DSS Files, page 13-13](#)
- [How to Edit DSS Files, page 13-16](#)
- [How to Import DSS Files, page 13-17](#)

## The Signature Editor Console

The Signature Editor writes log and error messages to the Signature Editor Console (in the Console view), when appropriate.

# Managing DSS Files

- Installing new signatures to an active service configuration is described in [Working with Protocol Packs, page 4-18](#).
- Working with signatures in the Service Configuration Editor is described in [Managing Protocol Signatures, page 7-46](#).
- Using **servconf**, the Server Configuration Utility, to apply signatures is described in [The Cisco SCA BB Service Configuration Utility, page 14-2](#).

The DSS file components, and the creation and editing of DSS files, are explained in the following sections.

## The DSS File Components

The DSS file components are displayed in the Script pane of the Signature Editor, in a tree structure. By selecting the appropriate node of the DSS component tree, you can define the properties associated with the node in the Property pane.

The DSS file components are described in the following sections.

- [The DSS File, page 13-2](#)
- [DSS Protocol List, page 13-3](#)
- [Information About DSS Protocols, page 13-3](#)
- [DSS Signatures, page 13-4](#)
- [DSS Deep Inspection Clauses, page 13-9](#)
- [DSS Deep Inspection Conditions, page 13-10](#)

## The DSS File

The DSS file name is the root node of the DSS file component tree.

When you select the root node, you can define the following properties for the DSS file:

- Script Name—Enter a meaningful name for this script.
- Script Description—Enter the reason for creating this script and describe its contents.
- Script Version (Major)
- Script Version (Minor)
- Script Build Number (Major)
- Script Build Number (Minor)
- Created for Application Version—Select from a list of predefined values.

Figure 13-1 shows the default values for the DSS file properties.

**Figure 13-1** Default Values for DSS File Properties

| Property                 | Value    |
|--------------------------|----------|
| Script Name              | MyScript |
| Script Description       |          |
| Script Version (Major)   | 1        |
| Script Version (Minor)   | 0        |
| Script Build no. (Major) | 1        |
| Script Build no. (Minor) | 0        |
| Created for App. Version | 3.1.0    |

The DSS file contains a single protocol list.

## DSS Protocol List

The protocol list has no properties to define. It contains all the protocols that are being added, modified, or enhanced.

## Information About DSS Protocols

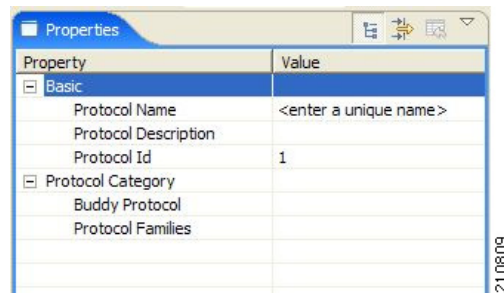
When you select a Protocol node in the DSS file component tree, you can define the following properties of the protocol:

- Basic:
  - Protocol Name—See [“Setting Protocol Name and ID”](#) section on page 13-4.
  - Protocol Description
  - Protocol ID—See [“Setting Protocol Name and ID”](#) section on page 13-4.
- Protocol Category:
  - Buddy Protocol—See [“The Buddy Protocol”](#) section on page 13-4.
  - Protocol Families—Assign the protocol to one or more protocol families:
    - P2P
    - SIP
    - VOIP
    - Worm

Associating a protocol with a protocol family allows reports about the family to include the new protocol.

Figure 13-2 shows the default values for the protocol properties.

**Figure 13-2** Default Values for the Protocol Properties



| Property             | Value                 |
|----------------------|-----------------------|
| Basic                |                       |
| Protocol Name        | <enter a unique name> |
| Protocol Description |                       |
| Protocol Id          | 1                     |
| Protocol Category    |                       |
| Buddy Protocol       |                       |
| Protocol Families    |                       |

Protocols contain signatures.

### Setting Protocol Name and ID

A DSS can include two types of protocols:

- A protocol new to Cisco SCA BB—The protocol is being defined in the DSS.
- A protocol that Cisco SCA BB already supports—The protocol identification is being enhanced or modified in the DSS.

Selecting a name and ID is different for the two cases:

- For a protocol new to Cisco SCA BB, the name must not match any of the protocol names that Cisco SCA BB already supports. To see a list of supported-protocol names, open the Protocol Settings dialog box in the Service Configuration Editor (see “[How to View Protocols](#)” section on page 7-23). Assign the protocol a unique ID in the range from 5000 to 9998.
- For an existing protocol, the protocol name and ID in the DSS must be identical to the protocol name and ID in the service configuration. Locate the name and ID in the Protocol Settings dialog box in the Service Configuration Editor (see “[How to View Protocols](#)” section on page 7-23).

### The Buddy Protocol

To simplify the configuration of new protocols added by a DSS, the DSS may specify a Buddy Protocol for a new protocol. If, when importing a DSS to a service configuration, the application encounters service elements referring to the Buddy Protocol, it automatically duplicates the set of service elements that use the Buddy Protocol and replaces all references to the Buddy Protocol with references to the new protocol. The association of the new protocol to services matches that of the Buddy Protocol.

## DSS Signatures

A protocol may contain as many different signatures as necessary.

Four different types of signatures may be added to a protocol:

- String Match Signatures
- Payload Length Signatures
- HTTP User Agent Signatures
- HTTP x-Header Signatures

Each of the four signature types tests different conditions against the first payload packet of the flows.

These signature types and their conditions are described in the following subsections.

String Match Signatures and Payload Length Signatures can contain deep inspection clauses. A signature whose first payload packet conditions are met accepts a flow if the conditions of any of its deep inspection clauses are also met.

### DSS String Match Signature

When you select a String Match Signature node in the DSS file component tree, you can define the following properties of the signature:

- Signature Name—A unique name
- Signature Description
- Signature ID—A value in the range from 0xC010000 to 0xC0100FF (decimal 201392128 to 201392383)
- First Payload Packet Conditions:
  - Fixed Size Byte String—(Display only) Shows the string formed by the next four fields:
    - [0]—Enter the ASCII code for the first byte of the string, or enter “\*” to indicate that any value is acceptable.
    - [1]—Enter the ASCII code for the second byte of the string, or enter “\*” to indicate that any value is acceptable.
    - [2]—Enter the ASCII code for the third byte of the string, or enter “\*” to indicate that any value is acceptable.
    - [3]—Enter the ASCII code for the fourth byte of the string, or enter “\*” to indicate that any value is acceptable.
  - String Position—The position of the Fixed Size Byte String in the packet. The position is the location of the first byte of the string, counting from the first byte in the packet. To match the string with the beginning of the packet, this value should be zero. The value must be an integer divisible by four.
  - Packet Direction—The initiating side of the first packet in the flow that has a payload. This field can have one of three values:
    - From Server
    - From Client
    - Don’t Care (either side)
  - Port Range—(Display only) The port range formed by the next two fields. The default value is the entire port range from 0 to 65535.
    - From Port—Lower bound of the port range (inclusive)
    - To Port—Upper bound of the port range (inclusive)
  - Check before PL—Toggles between the values **true** and **false**.

This field indicates whether to test the signature before or after the execution of the Cisco SCA BB built-in PL (Protocol Library) classification. Testing this signature before the execution of the built-in classification means that if the flow matches this signature, the PL classification is skipped. If this field is set to “false”, this signature is tested only if the PL classification fails to identify any of its supported protocol signatures.
  - Asymmetric Routing Classification Mode—This field indicates whether to test the signature depending on the state of the asymmetric routing classification mode. It can have one of three values:

- Don't Care—Signifies that this signature should be tested whether asymmetric routing classification mode is enabled or disabled.
- Disabled
- Enabled
- Flow Type—(Display only) This field shows to which flow types the condition applies (the condition may be applied to multiple types). It is ignored unless asymmetric routing classification mode is enabled.

The next four fields specify the flow type:

- Bidirectional—Toggles between the values **true** and **false**.
- Unidirectional Client Side—Toggles between the values **true** and **false**. Applies to TCP flows for which only packets from the client side have been detected.
- Unidirectional Server Side—Toggles between the values **true** and **false**. Applies to TCP flows for which only packets from the server side have been detected.
- Unknown (UDP)—Toggles between the values **true** and **false**. Applies to UDP flows for which packets from only one direction have been detected.



#### Caution

Set Check before PL to **true** only if the signature identifies the protocol according to the first payload packet only. If the signature also uses a Deep Inspection Condition that looks into later packets, and the signature does not match the flow, the PL classification is not performed properly.

Figure 13-3 shows the default values for the String Match Signature properties.

**Figure 13-3** Default Values for the String Match Signature Properties

| Property  | Value                 |
|---|-----------------------|
| Signature Name  | <enter a unique name> |
| Signature Description   |                       |
| Signature Id  | 0xC010000             |
| <input checked="" type="checkbox"/> First Payload Packet Conditions |                       |
| <input checked="" type="checkbox"/> Fixed Size Byte String          | abcd                  |
| [0]   | 97                    |
| [1]   | 98                    |
| [2]   | 99                    |
| [3]   | 100                   |
| String Position   | 0                     |
| Packet Direction  | Don't Care            |
| <input checked="" type="checkbox"/> Port Range                      | 0:65535               |
| From port   | 0                     |
| To port   | 65535                 |
| Check before PL   | false                 |
| Asymmetric Routing Classification Mode                              | Don't Care            |
| <input checked="" type="checkbox"/> Flow Type                       | Bidirectional         |
| Bidirectional   | true                  |
| Unidirectional Client Side  | false                 |
| Unidirectional Server Side  | false                 |
| Unknown (UDP)   | false                 |

A flow that matches the first payload packet conditions of a String Match Signature is then compared against the deep inspection conditions of the signature (see “DSS Deep Inspection Conditions” section on page 13-10).

### DSS Payload Length Signature

When you select a Payload Length Signature node in the DSS file component tree, you can define the following properties of the signature:

- Signature Name—A unique name
- Signature Description
- Signature ID—A value in the range from 0xC010000 to 0xC0100FF (decimal 201392128 to 201392383)
- First Payload Packet Conditions:
  - Packet Direction—The initiating side of the first packet in the flow that has a payload. This field can have one of three values:
    - From Server
    - From Client
    - Don't Care (either side)
  - Payload Length—The number of bytes in the payload packet.
  - Port Range—(Display only) The port range formed by the next two fields. The default value is the entire port range from 0 to 65535.
  - From Port—Lower bound of the port range (inclusive)
  - To Port—Upper bound of the port range (inclusive)
  - Check before PL—Toggles between the values **true** and **false**.

This field indicates whether to test the signature before or after the execution of the Cisco SCA BB built-in PL (Protocol Library) classification. Testing this signature before the execution of the built-in classification means that if the flow matches this signature, the PL classification is skipped. If this field is set to “false”, this signature is tested only if the PL classification fails to identify any of its supported protocol signatures.

- Asymmetric Routing Classification Mode—This field indicates whether to test the signature depending on the state of the asymmetric routing classification mode. It can have one of three values:
  - Don't Care—Signifies that this signature should be tested whether asymmetric routing classification mode is enabled or disabled.
  - Disabled
  - Enabled
- Flow Type—(Display only) This field shows to which flow types the condition applies (the condition may be applied to multiple types). It is ignored unless asymmetric routing classification mode is enabled.

The next four fields specify the flow type:

- Bidirectional—Toggles between the values **true** and **false**.
- Unidirectional Client Side—Toggles between the values **true** and **false**. Applies to TCP flows for which only packets from the client side have been detected.
- Unidirectional Server Side—Toggles between the values **true** and **false**. Applies to TCP flows for which only packets from the server side have been detected.
- Unknown (UDP)—Toggles between the values **true** and **false**. Applies to UDP flows for which packets from only one direction have been detected.

**Caution**

Set Check before PL to **true** only if the signature identifies the protocol according to the first payload packet only. If the signature also uses a Deep Inspection Condition that looks into later packets, and the signature does not match the flow, the PL classification is not performed properly.

Figure 13-4 shows the default values for the Payload Length Signature properties.

**Figure 13-4** Default Values for the Payload Length Signature Properties

| Property                               | Value                 |
|--|-----------------------|
| Signature Name                         | <enter a unique name> |
| Signature Description                  |                       |
| Signature Id                           | 0xC010000             |
| First Payload Packet Conditions        |                       |
| Packet Direction                       | Don't Care            |
| Payload Length                         | 1                     |
| Port Range                             | 0:65535               |
| From port                              | 0                     |
| To port                                | 65535                 |
| Check before PL                        | false                 |
| Asymmetric Routing Classification Mode | Don't Care            |
| Flow Type                              | Bidirectional         |
| Bidirectional                          | true                  |
| Unidirectional Client Side             | false                 |
| Unidirectional Server Side             | false                 |
| Unknown (UDP)                          | false                 |

A flow that matches the first payload packet conditions of a Payload Length Signature is then compared against the deep inspection conditions of the signature (see “DSS Deep Inspection Conditions” section on page 13-10).

### DSS HTTP User Agent Signature

When you select an HTTP User Agent Signature node in the DSS file component tree, you can define the following properties of the signature:

- Signature Name—A unique name
- Signature Description
- Signature ID—A value in the range from 0xC010000 to 0xC0100FF (decimal 201392128 to 201392383)
- Conditions:
  - User Agent—The value of the User Agent field in the HTTP header

Figure 13-5 shows the default values for the HTTP User Agent signature properties.



**Figure 13-5** Default Values for the HTTP User Agent Signature Properties

| Property              | Value                 |
|-----------------------|-----------------------|
| Signature Name        | <enter a unique name> |
| Signature Description |                       |
| Signature Id          | 0xC010000             |
| Conditions            |                       |
| User Agent            | <user agent>          |

**DSS HTTP x-Header Signature**

When you select an HTTP x-Header Signature node in the DSS file component tree, you can define the following properties of the signature:

- Signature Name—A unique name
- Signature Description
- Signature ID—A value in the range from 0xC010000 to 0xC0100FF (decimal 201392128 to 201392383)
- Conditions:
  - x-Header Field Name—A name of a field in the x-Header of the HTTP header

Figure 13-6 shows the default values for the DSS file properties.

**Figure 13-6** Default Values for the DSS File Properties

| Property              | Value                 |
|-----------------------|-----------------------|
| Signature Name        | <enter a unique name> |
| Signature Description |                       |
| Signature Id          | 0xC010000             |
| Conditions            |                       |
| x-Header Field Name   | <field name>          |

**DSS Deep Inspection Clauses**

A deep inspection clause is a conjunctive clause of deep inspection conditions—a signature accepts a flow *only* if all conditions in a clause are met.

**Note**

If a signature has multiple deep inspection clauses, the clauses (and the deep inspection conditions making up each clause) are tested in an order based on the value of the Packet Number property of the deep inspection conditions.

After the first payload packet is accepted by the first payload packet conditions, the clause containing the condition with the lowest Packet Number is tested. The other conditions in this clause are checked in ascending Packet Number order. Thus, the Packet Number of any condition in a clause cannot be less than the largest Packet Number in the clause it succeeds.

## DSS Deep Inspection Conditions

A deep inspection condition is a set of conditions that are checked against flows that pass the first payload packet conditions screening of String Match Signatures or Payload Length Signatures.

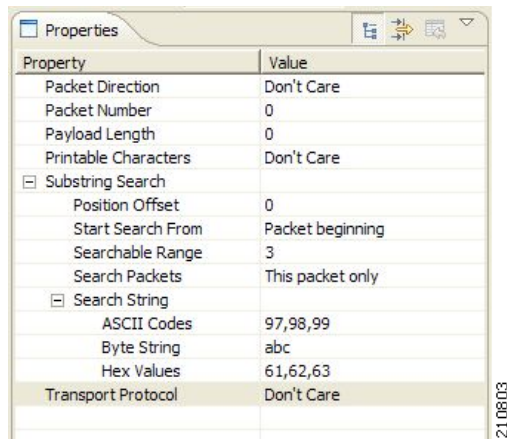
When you select a Deep Inspection Condition node in the DSS file component tree, you can define the following properties of the deep inspection condition:

- Packet Direction—The initiating side of the first packet in the flow that has a payload. This field can have one of three values:
  - From Server
  - From Client
  - Don't Care (either side)
- Packet Number—The number of the packet in the flow. The payload packets are numbered from zero; packets are counted in both directions.
- Payload Length—The length of the packet in bytes. Enter zero to indicate that any value is acceptable.
- Printable Characters—Test if the inspected packet contains only printable characters. This field can have one of three values:
  - Printable Characters Only
  - At Least One Non-Printable
  - Don't Care
- Substring Search—Match a search string with a specific location in the packet. Leave the Search String fields empty if this condition is irrelevant.
  - Position Offset—The position from which to start searching for the search string in the packet. The offset is relative to the location specified in the Start Search From field.
  - Start Search From—This field can have one of two values:
    - Packet beginning
    - Last matchLast match means that the search for this search string starts where the last search match ended. The last match may be from a previous substring search or from the last string-based first payload packet condition.
  - Searchable Range—Search in this number of bytes for the search string.
  - Search Packets—This field can have one of two values:
    - This packet only
    - Multiple packetsMultiple Packets means that the search may span across packets, as long as the overall number of bytes is less than the number specified in the Searchable Range field.
  - Search String—Enter the search string in one of the following three fields (the other two fields are updated automatically):
    - ASCII Codes—Enter the ASCII codes for the characters of the search string. Separate each code by a comma.
    - Byte String—Enter the actual search string.
    - Hex Values—Enter the hexadecimal values of the ASCII codes for the characters of the search string. Separate each code by a comma.

- Transport Protocol—This field can have one of three values:
  - TCP
  - UDP
  - Don't Care (either TCP or UDP)

Figure 13-7 shows the default values for the deep inspection condition properties.

**Figure 13-7** Default Values for the Deep Inspection Condition Properties




| Property   | Value            |
|--|------------------|
| Packet Direction                                     | Don't Care       |
| Packet Number  | 0                |
| Payload Length                                       | 0                |
| Printable Characters                                 | Don't Care       |
| <input checked="" type="checkbox"/> Substring Search |                  |
| Position Offset                                      | 0                |
| Start Search From                                    | Packet beginning |
| Searchable Range                                     | 3                |
| Search Packets                                       | This packet only |
| <input checked="" type="checkbox"/> Search String    |                  |
| ASCII Codes  | 97,98,99         |
| Byte String  | abc              |
| Hex Values   | 61,62,63         |
| Transport Protocol                                   | Don't Care       |

The structure of deep inspection conditions is the same for String Match Signatures and Payload Length Signatures.

# How to Create DSS Files

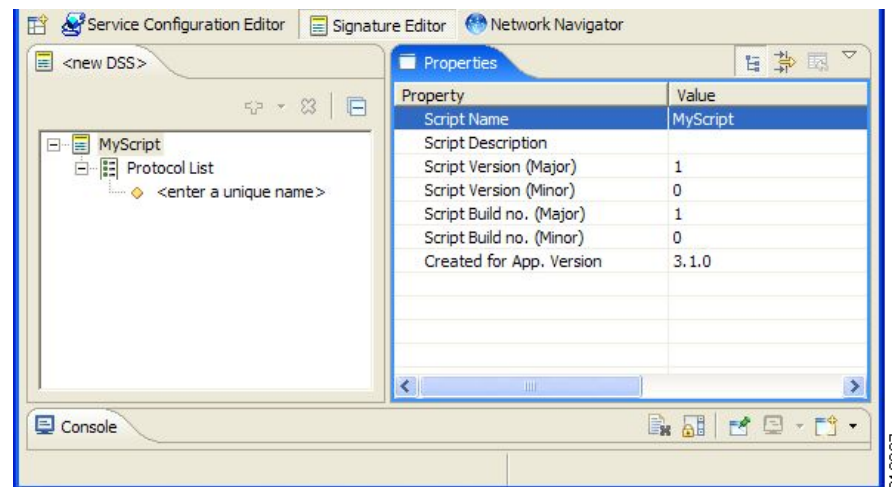
If you have a DSS file open in the Signature Editor, save it before you create a new DSS file. All unsaved changes are lost.

**Step 1** From the toolbar, click the **Create a New DSS File** (  ) icon.

A DSS component tree containing a DSS File node, a Protocol List node, and a Protocol node, is displayed in the Script view.

The default properties of the new DSS file are displayed in the Properties view ([Figure 13-8](#)).

**Figure 13-8** Properties Tab



**Step 2** Edit the DSS file properties.

For an explanation of the properties, see [“The DSS File” section on page 13-2](#).

**Step 3** Click the Protocol node.

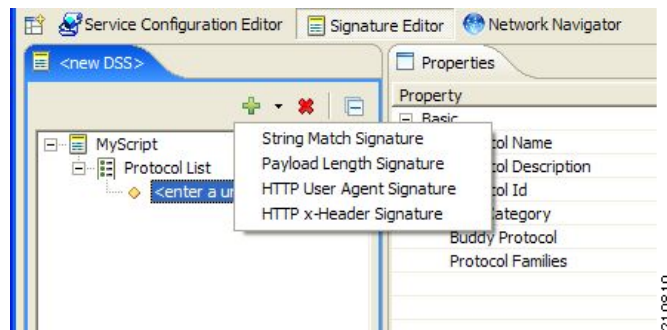
The protocol properties appear in the Properties view (Figure 13-9).

**Step 4** Edit the protocol properties.

For an explanation of the properties, see “[Information About DSS Protocols](#)” section on page 13-3.

**Step 5** Click the drop-down arrow next to the **Add** (+) icon.

**Figure 13-9 Protocol Properties**

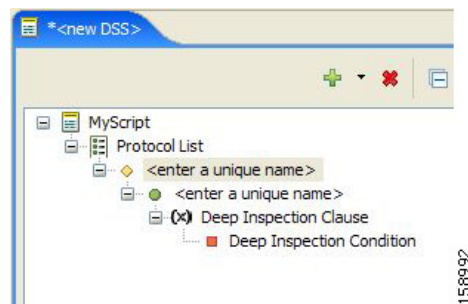


**Step 6** From the drop-down menu that appears, select a signature type.

A Signature node is added under the Protocol node.

If you selected a String Match Signature or a Payload Length Signature, a Deep Inspection Clause node and a Deep Inspection Condition node are also added (Figure 13-10).

**Figure 13-10 Protocol List Information**



**Step 7** Click the Signature node.

The signature properties appear in the Properties view.

**Step 8** Edit the signature properties.


For an explanation of the properties, see “[DSS Signatures](#)” section on page 13-4

**Step 9** If you selected a String Match Signature or a Payload Length Signature, click the Deep Inspection Condition node to edit the deep inspection condition properties.

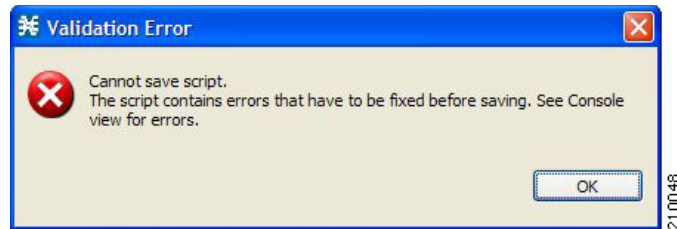
The deep inspection condition properties appear in the Properties view.


For an explanation of the properties, see “[DSS Deep Inspection Conditions](#)” section on page 13-10.

**Step 10** Add additional deep inspection conditions, deep inspection clauses, signatures, and protocols as needed.

- Step 11** From the toolbar, click the **Save** (  ) icon.
- If there are duplicate protocol names or protocol IDs, a Validation Error message appears (Figure 13-11).

**Figure 13-11** Validation Error



Click **OK**, remove the duplication, and then click the **Save** (  ) icon again.

A Save As dialog box appears.

- Step 12** Browse to the folder where you want to save the new DSS file.
- Step 13** In the File name field, enter an appropriate name for the DSS file.
- Step 14** Click **Save**.

The Save As dialog box closes.

The DSS file is saved.

---


# How to Edit DSS Files

You can edit an existing DSS file, and add new protocols, or modify or delete existing protocols.

**Caution**

If you have a DSS file open in the Signature Editor, save it before you open a different DSS file. All unsaved changes are lost.

---

**Step 1** From the toolbar, click the **Open a DSS File** () icon.

An Open dialog box appears.

**Step 2** Browse to the DSS file that you want to edit.

**Step 3** Click **Open**.

The Open dialog box closes.


The DSS Component tree of the selected file is displayed in the Script view.

The DSS File node is selected, and the properties of the DSS file are displayed in the Properties view.

**Step 4** Add, edit, or delete DSS file components.

See the subsections of “[The DSS File Components](#)” section on page 13-2 for an explanation of the properties of the different components.

**Step 5** Save the modified DSS file.

- To overwrite the current DSS file with the changes you have made:
  - From the toolbar, click the **Save** () icon.

The changes to the DSS file are saved.

- To save the modified DSS file with a new name:

**1.** Choose **File > Save As**.

A Save As dialog box appears.

**2.** Browse to the folder where you want to save the new DSS file.

**3.** In the File name field, enter an appropriate name for the DSS file.

**4.** Click **Save**.

The Save As dialog box closes.

The modified DSS file is saved with the new name.

---



# How to Import DSS Files

You can import DSS files into the file you are currently editing.

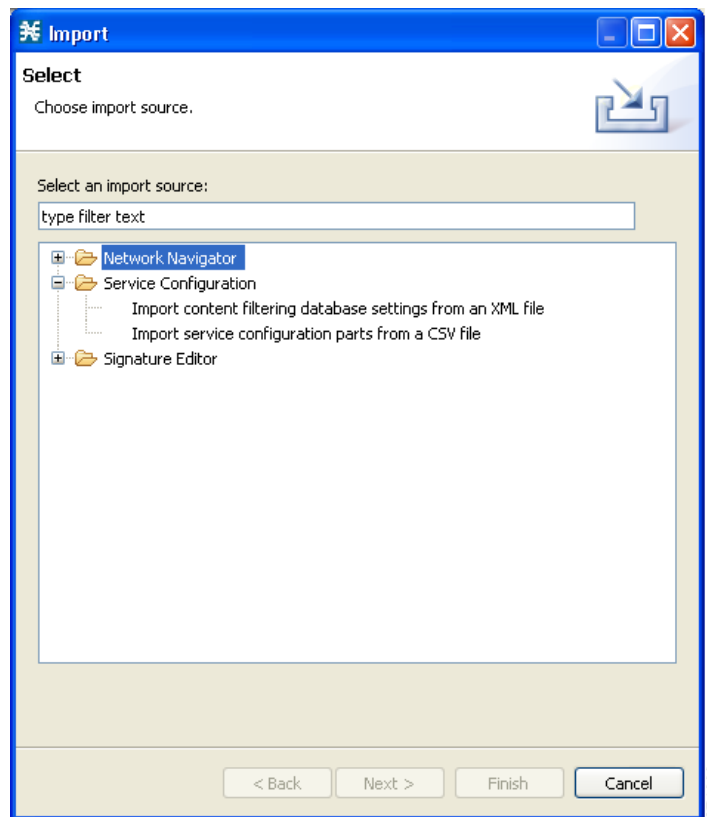


## Note

Importing signatures may create duplication of protocol names or protocol IDs.

- Step 1** From the Console main menu, choose **File > Import**.  
The Import dialog box appears (Figure 13-12).

**Figure 13-12** *Import*

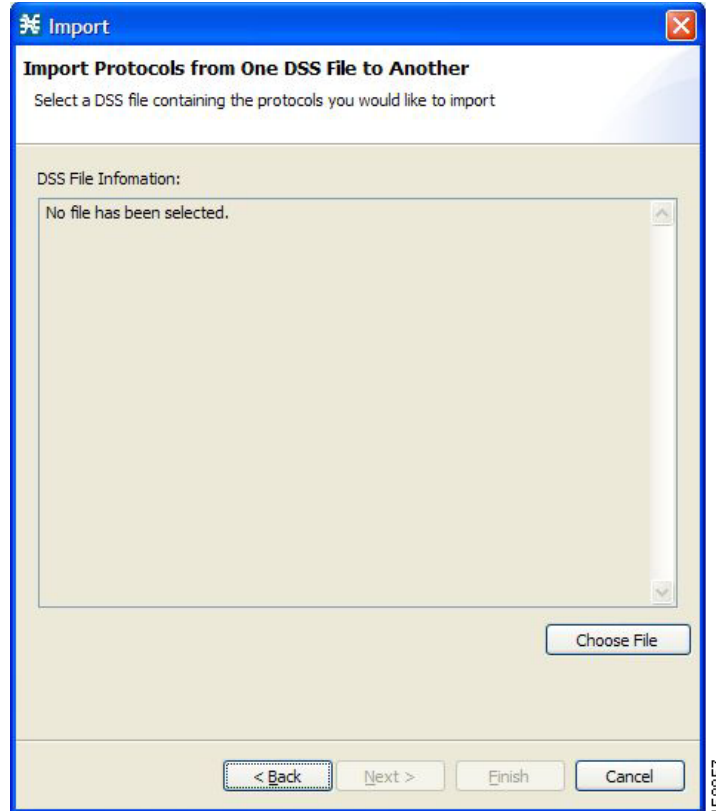


- Step 2** From the import source list, select **Import protocols from one DSS file to another DSS**.

**Step 3** Click **Next**.

The second screen of the Import dialog box opens (Figure 13-13).

**Figure 13-13** Import Protocols from One DSS File to Another



**Step 4** Click **Choose File**.

An Open dialog box appears.

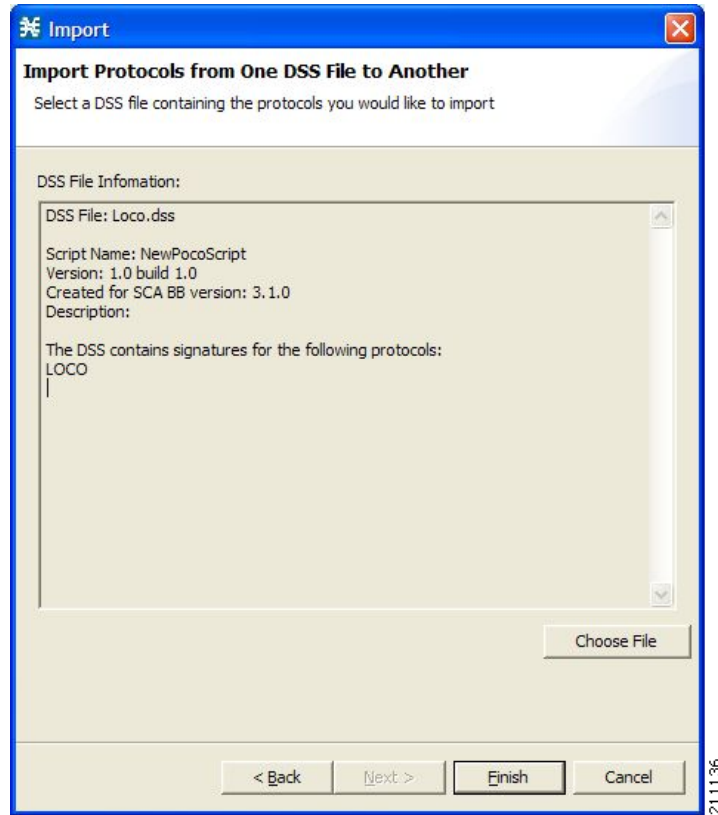
**Step 5** Browse to the DSS file to import.

**Step 6** Click **Open**.

The Open dialog box closes.

Information about the DSS file that you have chosen is displayed in the DSS File Information area (Figure 13-14).

**Figure 13-14** Import Protocols from One DSS File to Another

**Step 7** Click **Finish**.

The Import dialog box closes.

The content of the selected DSS file is imported into the Signature Editor.

