http://www.sapninja.com

Splunk for SAP

version 1.0

by Jim Cooke

Installation Manual

15 August 2011

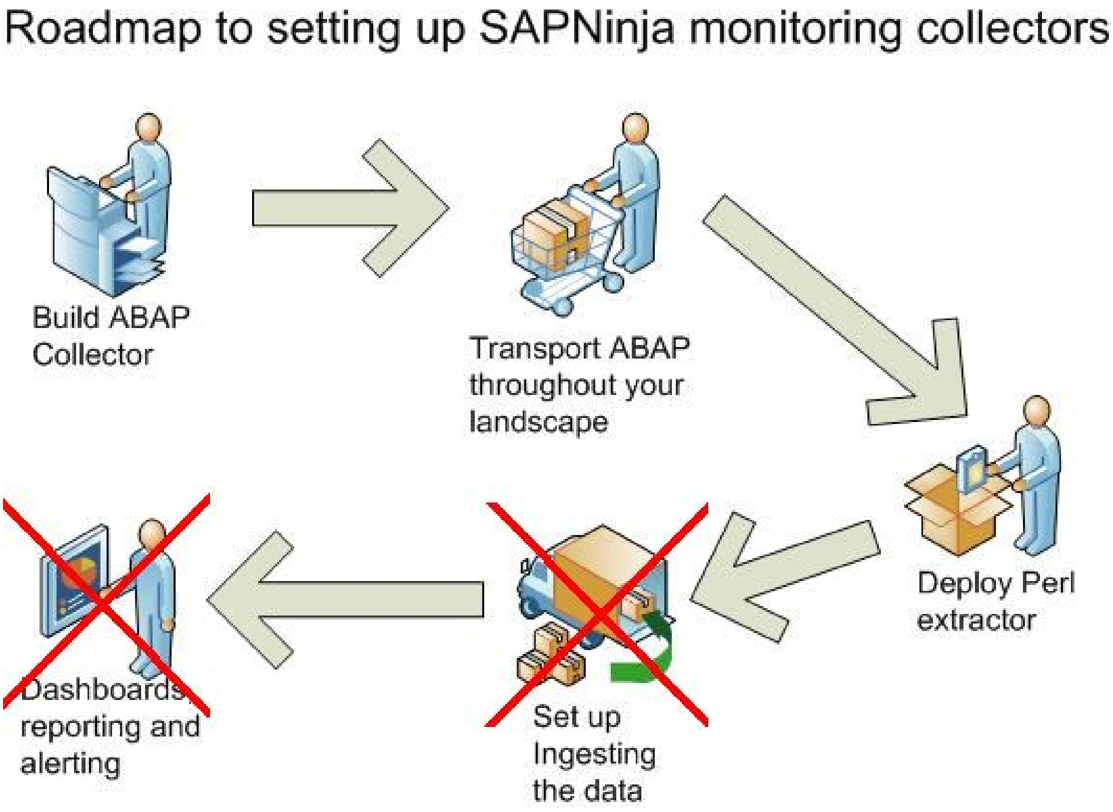
The latest version of this document can be found at
http://code.google.com/p/sapninja/source/browse/trunk/splunk/SplunkForSAP.doc

PO BOX 712
RANDWICK, NSW 2031
AUSTRALIA
sapslunk@gmail.com

Table of Contents

Roadmap

In terms of the roadmap to setting up information collectors for Splunk, this document is broken up into three sections, corresponding to the first three sections of the roadmap. The final two sections involve the setup of your reporting solution and are beyond the scope of this document (and of the http://www.sapninja.com website for that matter).


Roadmap to setting up SAPNinja monitoring collectors
Build ABAP Collector
Transport ABAP throughout your landscape
Deploy Perl extractor
Set up Ingesting the data
Dashboards, reporting and alerting

Section 1 – Build Collector Framework

The ABAP components consist of the following:

1. Staging tables to hold data that has not yet been ingested by Splunk
2. A collector program to gather data and populate the staging tables
3. A remote-enabled function which can be called by Splunk to supply data that has not yet been ingested.

Development Objects

These tasks can be assigned to any ABAP programmer, who will be familiar with the concepts.

Development Package

All objects created from now on should be assigned to package ***ZPLUNK***, if prompted. Start by creating this development package.

Start transaction **SE80**

From the top drop-down on the left-hand pane, choose

Object type = Package
Enter Object Name = ZPLUNK

Click ***Enter***

Answer ***Yes*** to create the object

Enter the following values for the object properties:

Property	Value
Package	ZPLUNK
Short Description	Splunk Collector - Development Package
Application Component	BC
Software Component	HOME
Transport Layer	<Your Choice>
Package Type	Not a main package

Click ***Enter***

Function Group

There will be a few functions created for the Splunk collector, including the main extraction function as well as maintenance screen functions. They need to belong to a function group. We will create a new function group called ***ZPLUNK_FGRP***.

Start transaction **SE80**

Open Package ***ZPLUNK***

Right-click on ***ZPLUNK*** and choose ***Create*** → ***Function Group***

Function Group = ZPLUNK_FGRP
Short Text = Splunk Collector – Function Group

Click ***Save***

Assign to package ***ZPLUNK*** when prompted

Data Types

There are many “standard” SAP data types that could fill this function, but it is better to create a set of custom data types so that we can be sure they will exist on all types of systems that we wish to use our collector on. For example some of the “standard” types might exist in an ERP installation, but not Solution Manager or, vice versa.

For each data type listed in the table below, repeat the following steps:

- Start transaction **SE11**
- Click on the ***Data Type*** radio button
- Enter the ***name*** from the “name” column below and Click ***Create***
- Select the ***Data Element*** radio button and Click ***Enter***
- At the next screen, always select the ***Predefined Type*** radio button
- Assign to the ***ZPLUNK*** package as mentioned before

Name	Type	Short Description	Data Type	Len	Label Name
ZPLUNK_TXT1	data element	Text - 1 character	CHAR	1	TEXT1
ZPLUNK_TXT2	data element	Text - 2 characters	CHAR	2	TEXT2
ZPLUNK_TXT3	data element	Text - 3 characters	CHAR	3	TEXT3
ZPLUNK_TXT4	data element	Text - 4 characters	CHAR	4	TEXT4
ZPLUNK_TXT12	data element	Text - 12 characters	CHAR	12	TEXT4
ZPLUNK_TXT20	data element	Text - 20 characters	CHAR	20	TEXT20
ZPLUNK_TXT40	data element	Text - 40 characters	CHAR	40	TEXT40
ZPLUNK_TXT60	data element	Text - 60 characters	CHAR	60	TEXT60
ZPLUNK_TXT120	data element	Text - 120 characters	CHAR	120	TEXT120
ZPLUNK_TXT255	data element	Text - 255 characters	CHAR	255	TEXT255
ZPLUNK_DATE	data element	Date	DATS	8	DATE
ZPLUNK_TIME	data element	Time	TIMS	6	TIME
ZPLUNK_DEC24	data element	Decimal	DEC	24	DEC24

Tables

The collector relies on several staging tables to store the data, as well as tables which control collector activities.

For each data type listed in the table below, repeat the following steps:

- Start transaction **SE11**
- Click on the ***Database Table*** radio button
- Enter the ***name*** from each sub-section below and Click ***Create***

ZPLUNK_CCMS_A

ATTRIBUTES

Short Description = ***Splunk Collector - CCMS Alerts***

DELIVERY AND MAINTENANCE

Delivery Class = ***L***
Data Browser/Table View Maint = ***Display/Maintenance Allowed***

FIELDS

Name	Key	Data Element
SAMPLEDATE	X	ZPLUNK_DATE
SAMPLETIME	X	ZPLUNK_TIME
HOST	X	ZPLUNK_TXT40
SYSNAME	X	ZPLUNK_TXT3
SYSNR	X	ZPLUNK_TXT2
MTECLASS	X	ZPLUNK_TXT120
SENT		ZPLUNK_TXT1
MSG		ZPLUNK_TXT255
SEVERITY		ZPLUNK_TXT1

EXTRAS → ENHANCEMENT CATEGORY

Category = ***Cannot be enhanced***

TECHNICAL SETTINGS

Data Class = ***USER***
Size Category = ***2***
Buffering = ***Buffering not allowed***

ZPLUNK_CCMS_C

ATTRIBUTES

Short Description = ***Splunk Collector - CCMS MTE Status***

DELIVERY AND MAINTENANCE

Delivery Class = ***L***

Data Browser/Table View Maint = ***Display/Maintenance Allowed***

FIELDS

Name	Key	Data Element
SAMPLEDATE	X	ZPLUNK_DATE
SAMPLETIME	X	ZPLUNK_TIME
HOST	X	ZPLUNK_TXT40
SYSNAME	X	ZPLUNK_TXT3
SYSNR	X	ZPLUNK_TXT2
MTECLASS	X	ZPLUNK_TXT120
SENT		ZPLUNK_TXT120
MSG		ZPLUNK_TXT255
SEVERITY		ZPLUNK_TXT1
OBSERVATION		INT4
OBSERVATION_UOM		ZPLUNK_TXT4

EXTRAS → ENHANCEMENT CATEGORY

Category = ***Cannot be enhanced***

TECHNICAL SETTINGS

Data Class = ***USER***

Size Category = ***3***

Buffering = ***Buffering not allowed***

ZPLUNK_CCMS_MTES

ATTRIBUTES

Short Description = ***Splunk Collector - CCMS MTE classes for current status***

DELIVERY AND MAINTENANCE

Delivery Class = ***L***
Data Browser/Table View Maint = ***Display/Maintenance Allowed***

FIELDS

Name	Key	Data Element
MTECLASS	X	ZPLUNK_TXT40
COLL_FREQ_MINS		INT4
THRESHOLD_ACTIVE		ZPLUNK_TXT1

EXTRAS → ENHANCEMENT CATEGORY

Category = ***Cannot be enhanced***

TECHNICAL SETTINGS

Data Class = ***USER***
Size Category = ***0***
Buffering = ***Buffering switched on***
Buffering Type = ***Fully Buffered***

ZPLUNK_PARAMS

ATTRIBUTES

Short Description = ***Splunk Collector – Application Parameters***

DELIVERY AND MAINTENANCE

Delivery Class = ***L***
Data Browser/Table View Maint = ***Display/Maintenance Allowed***

FIELDS

Name	Key	Data Element
PARAM	X	ZPLUNK_TXT60
VALUE		ZPLUNK_TXT60

EXTRAS → ENHANCEMENT CATEGORY

Category = ***Cannot be enhanced***

TECHNICAL SETTINGS

Data Class = ***USER***
Size Category = ***0***
Buffering = ***Buffering switched on***
Buffering Type = ***Fully Buffered***

TABLE MAINTENANCE

You generate a table maintenance program for this table. From **SE11**:

- *Utilities → Table Maintenance Generator*

Authorization Group	&NC&
Maintenance Type	One step
Function group	ZPLUNK_FGRP

- Click on ***Find Scr. Number(s)*** button
- Select ***Propose screen number(s)*** radio button
- Click ***Enter***
- Click ***Create***

ZPLUNK_SM04MEM

ATTRIBUTES

Short Description = ***Splunk Collector – SM04 Memory Usage***

DELIVERY AND MAINTENANCE

Delivery Class = ***L***
Data Browser/Table View Maint = ***Display/Maintenance Allowed***

FIELDS

Name	Key	Data Element
SAMPLEDATE	X	ZPLUNK_DATE
SAMPLETIME	X	ZPLUNK_TIME
HOST	X	ZPLUNK_TXT40
SYSNAME	X	ZPLUNK_TXT3
SYSNR	X	ZPLUNK_TXT2
IDX	X	INT2
SENT		ZPLUNK_TXT1
MANDT		ZPLUNK_TXT3
BNAME		ZPLUNK_TXT12
TCODE		ZPLUNK_TXT20
PAGEMEM		ZPLUNK_DEC24
ROLLMEM		ZPLUNK_DEC24
EXTENDEDMEM		ZPLUNK_DEC24
PRIVMEM		ZPLUNK_DEC24

EXTRAS → ENHANCEMENT CATEGORY

Category = ***Cannot be enhanced***

TECHNICAL SETTINGS

Data Class = ***USER***
Size Category = ***3***
Buffering = ***Buffering not allowed***

ZPLUNK_SM04USR

ATTRIBUTES

Short Description = ***Splunk Collector – SM04 User List***

DELIVERY AND MAINTENANCE

Delivery Class = ***L***

Data Browser/Table View Maint = ***Display/Maintenance Allowed***

FIELDS

Name	Key	Data Element
SAMPLEDATE	X	ZPLUNK_DATE
SAMPLETIME	X	ZPLUNK_TIME
HOST	X	ZPLUNK_TXT40
SYSNAME	X	ZPLUNK_TXT3
SYSNR	X	ZPLUNK_TXT2
IDX	X	INT2
SENT		ZPLUNK_TXT1
MANDT		ZPLUNK_TXT3
BNAME		ZPLUNK_TXT12
USTYP		ZPLUNK_TXT12
TCODE		ZPLUNK_TXT20
LASTACTION		ZPLUNK_TIME
TERM		ZPLUNK_TXT40
IPADDR		ZPLUNK_TXT40

EXTRAS → ENHANCEMENT CATEGORY

Category = ***Cannot be enhanced***

TECHNICAL SETTINGS

Data Class = ***USER***

Size Category = ***3***

Buffering = ***Buffering not allowed***

ZPLUNK_SM21

ATTRIBUTES

Short Description = ***Splunk Collector – SM21 System Log***

DELIVERY AND MAINTENANCE

Delivery Class = ***L***
Data Browser/Table View Maint = ***Display/Maintenance Allowed***

FIELDS

Name	Key	Data Element
SAMPLEDATE	X	ZPLUNK_DATE
SAMPLETIME	X	ZPLUNK_TIME
HOST	X	ZPLUNK_TXT40
SYSNAME	X	ZPLUNK_TXT3
SYSNR	X	ZPLUNK_TXT2
ENTRY	X	ZPLUNK_TXT12
ENTRYTYPE		ZPLUNK_TXT1
MSGNO		ZPLUNK_TXT3
MANDT		ZPLUNK_TXT3
BNAME		ZPLUNK_TXT12
TERM		ZPLUNK_TXT10
MODE		ZPLUNK_TXT1
PID		ZPLUNK_TXT12
TASKNO		ZPLUNK_TXT12
TASKTYPE		ZPLUNK_TXT2
TCODE		ZPLUNK_TXT20
REPNAME		ZPLUNK_TXT60
MSGNO		ZPLUNK_TXT120

EXTRAS → ENHANCEMENT CATEGORY

Category = ***Cannot be enhanced***

TECHNICAL SETTINGS

Data Class = ***USER***
Size Category = ***3***
Buffering = ***Buffering not allowed***

ZPLUNK_SM50

ATTRIBUTES

Short Description = ***Splunk Collector – SM50 Work Processes***

DELIVERY AND MAINTENANCE

Delivery Class = ***L***
Data Browser/Table View Maint = ***Display/Maintenance Allowed***

FIELDS

Name	Key	Data Element
SAMPLEDATE	X	ZPLUNK_DATE
SAMPLETIME	X	ZPLUNK_TIME
HOST	X	ZPLUNK_TXT40
SYSNAME	X	ZPLUNK_TXT3
SYSNR	X	ZPLUNK_TXT2
WP_NO	X	ZPLUNK_TXT2
SENT		ZPLUNK_TXT1
WP_TYPE		ZPLUNK_TXT3
WP_PID		ZPLUNK_TXT12
WP_STATUS		ZPLUNK_TXT12
WP_DUMPS		ZPLUNK_TXT2
WP_MANDT		ZPLUNK_TXT3
WP_BNAME		ZPLUNK_TXT12
WP_REPORT		ZPLUNK_TXT40
WP_ELTIME		ZPLUNK_TXT12
WP_ACTION		ZPLUNK_TXT40
WP_TABLE		ZPLUNK_TXT40

EXTRAS → ENHANCEMENT CATEGORY

Category = ***Cannot be enhanced***

TECHNICAL SETTINGS

Data Class = ***USER***
Size Category = ***3***
Buffering = ***Buffering not allowed***

ZPLUNK_SPOOL

ATTRIBUTES

Short Description = ***Splunk Collector – SP01 Successful Print Jobs***

DELIVERY AND MAINTENANCE

Delivery Class = ***L***
Data Browser/Table View Maint = ***Display/Maintenance Allowed***

FIELDS

Name	Key	Data Element
SAMPLEDATE	X	ZPLUNK_DATE
SAMPLETIME	X	ZPLUNK_TIME
HOST	X	ZPLUNK_TXT40
SYSNAME	X	ZPLUNK_TXT3
SYSNR	X	ZPLUNK_TXT2
SPOOLID	X	INT4
SENT		ZPLUNK_TXT1
MANDT		ZPLUNK_TXT3
BNAME		ZPLUNK_TXT12
PADEST		ZPLUNK_TXT4
LNAME		ZPLUNK_TXT40
PAGESPRINTED		INT2
BYTESOUT		INT4
OUTPUTTYPE		ZPLUNK_TXT12

EXTRAS → ENHANCEMENT CATEGORY

Category = ***Cannot be enhanced***

TECHNICAL SETTINGS

Data Class = ***USER***
Size Category = ***3***
Buffering = ***Buffering not allowed***

ZPLUNK_ST03N

ATTRIBUTES

Short Description = ***Splunk Collector – ST03N Workload Overview***

DELIVERY AND MAINTENANCE

Delivery Class = ***L***
Data Browser/Table View Maint = ***Display/Maintenance Allowed***

FIELDS

Name	Key	Data Element
SAMPLEDATE	X	ZPLUNK_DATE
SAMPLETIME	X	ZPLUNK_TIME
HOST	X	ZPLUNK_TXT40
SYSNAME	X	ZPLUNK_TXT3
SYSNR	X	ZPLUNK_TXT2
TASKTYPE	X	ZPLUNK_TXT60
SENT		ZPLUNK_TXT1
STEPS		ZPLUNK_DEC24
RESPTI		ZPLUNK_DEC24
PROCTI		ZPLUNK_DEC24
CPUTI		ZPLUNK_DEC24
DBTI		ZPLUNK_DEC24
QUEUETI		ZPLUNK_DEC24
LOADGENTI		ZPLUNK_DEC24
LOCKTI		ZPLUNK_DEC24
CPICTI		ZPLUNK_DEC24
GUINETTIME		ZPLUNK_DEC24
GUITIME		ZPLUNK_DEC24
GUICNT		ZPLUNK_DEC24
BYTES		ZPLUNK_DEC24
ROLLINSTEP		ZPLUNK_DEC24
ROLLINTI		ZPLUNK_DEC24
ROLLOUTTI		ZPLUNK_DEC24
ROLLOUTCNT		ZPLUNK_DEC24
ROLLWAITTI		ZPLUNK_DEC24
ROLLSTEP		ZPLUNK_DEC24
VMC_CALL_COUNT		ZPLUNK_DEC24
VMC_CPU_TIME		ZPLUNK_DEC24
VMC_ELAP_TIME		ZPLUNK_DEC24
PHYREADCNT		ZPLUNK_DEC24
CHNGCNT		ZPLUNK_DEC24
READDIRBUF		ZPLUNK_DEC24
PHYCHNGREC		ZPLUNK_DEC24
READSEQCNT		ZPLUNK_DEC24

EXTRAS → ENHANCEMENT CATEGORY

Category = ***Cannot be enhanced***

TECHNICAL SETTINGS

Data Class = ***USER***
Size Category = ***3***
Buffering = ***Buffering not allowed***

ZPLUNK_ST03N_FE

ATTRIBUTES

Short Description = ***Splunk Collector – ST03N Frontend Usage***

DELIVERY AND MAINTENANCE

Delivery Class = ***L***
Data Browser/Table View Maint = ***Display/Maintenance Allowed***

FIELDS

Name	Key	Data Element
SAMPLEDATE	X	ZPLUNK_DATE
SAMPLETIME	X	ZPLUNK_TIME
HOST	X	ZPLUNK_TXT40
SYSNAME	X	ZPLUNK_TXT3
SYSNR	X	ZPLUNK_TXT2
PHOST	X	ZPLUNK_TXT60
SENT		ZPLUNK_TXT1
STEPS		ZPLUNK_DEC24
BYTESIN		ZPLUNK_DEC24
BYTESOUT		ZPLUNK_DEC24
GUITIME		ZPLUNK_DEC24
GUICNT		ZPLUNK_DEC24
GUINETTIME		ZPLUNK_DEC24

EXTRAS → ENHANCEMENT CATEGORY

Category = ***Cannot be enhanced***

TECHNICAL SETTINGS

Data Class = ***USER***
Size Category = ***3***
Buffering = ***Buffering not allowed***

ZPLUNK_ST03N_TRN

ATTRIBUTES

Short Description = ***Splunk Collector – ST03N User Transactions***

DELIVERY AND MAINTENANCE

Delivery Class = ***L***
Data Browser/Table View Maint = ***Display/Maintenance Allowed***

FIELDS

Name	Key	Data Element
SAMPLEDATE	X	ZPLUNK_DATE
SAMPLETIME	X	ZPLUNK_TIME
HOST	X	ZPLUNK_TXT40
SYSNAME	X	ZPLUNK_TXT3
SYSNR	X	ZPLUNK_TXT2
TASKTYPE	X	ZPLUNK_TXT60
BNAME	X	ZPLUNK_TXT12
EXECUTED		ZPLUNK_TXT120
SENT		ZPLUNK_TXT1
CALLTYPE		ZPLUNK_TXT1
STEPS		ZPLUNK_DEC24
RESPTI		ZPLUNK_DEC24
PROCTI		ZPLUNK_DEC24
CPUTI		ZPLUNK_DEC24
QUEUETI		ZPLUNK_DEC24
ROLLWAITTI		ZPLUNK_DEC24
GUITIME		ZPLUNK_DEC24
GUICNT		ZPLUNK_DEC24
GUINETTIME		ZPLUNK_DEC24
DBTI		ZPLUNK_DEC24
READDIRTI		ZPLUNK_DEC24
READSEQTI		ZPLUNK_DEC24
CHNGTI		ZPLUNK_DEC24

EXTRAS → ENHANCEMENT CATEGORY

Category = ***Cannot be enhanced***

TECHNICAL SETTINGS

Data Class = ***USER***
Size Category = ***3***
Buffering = ***Buffering not allowed***

ABAP Classes

The following ABAP class is used by the Splunk collector to perform some generic functions and formatting.

- Execute transaction ***SE24***
- Enter *Object type* = ***ZPLUNK_COMMON***
- Click ***Create***
- Choose the ***Class*** radio button and Click ***Enter***
- Set *Instantiation* = ***Public***
- Click ***Enter***
- Set *Package* = ***ZPLUNK***
- Click ***Enter***
- Open up the ***Methods*** tab and perform the actions listed below for each method

KEYPAIRT method

METHOD DESCRIPTOR

Method	Level	Visibility	Description
KEYPAIRT	Static	Public	Generate a text keypair with text input

METHOD PARAMETERS

Click the *Parameters* button to define these

Parameter	Type	Pass Val	Method	Type	Description
KEY	Importing	X	Type	ZPLUNK_TXT255	Key name
VALUE	Importing	X	Type	ZPLUNK_TXT255	Key value
RESULT	Exporting	X	Type	ZPLUNK_TXT255	Keypair

METHOD CODE

- Click ***Save***
- Click ***Back*** (green arrow)
- Double-click KEYPAIRT
- From menu *Utilities* → *More Utilities* → *Upload/Download* → *Upload*
- Import code from *meth_keypairt.txt* at the link

http://code.google.com/p/sapninja/source/browse/trunk/splunk/meth_keypairt.txt

- Click ***Save***

CHECKLOCK method

METHOD DESCRIPTOR

Method	Level	Visibility	Description
CHECKLOCK	Static	Public	Check dataset lock record

METHOD PARAMETERS

Click the *Parameters* button to define these

Parameter	Type	Pass Val	Method	Type	Description
DATASET	Importing	X	Type	ZPLUNK_TXT255	Splunk Dataset
EXPIRY_DAET	Importing	X	Type	ZPLUNK_DATE	Lock expiry date
EXPIRY_TIME	Importing	X	Type	ZPLUNK_TIME	Lock expiry time
RESULT	Exporting	X	Type	ZPLUNK_TXT255	Result String

METHOD CODE

Click ***Save***

Click ***Back*** (green arrow)

Double-click KEYPAIRT

From menu *Utilities* ⟶ *More Utilities* ⟶ *Upload/Download* ⟶ *Upload*
Import code from *meth_checklock.txt* at the link

http://code.google.com/p/sapninja/source/browse/trunk/splunk/meth_checklock.txt

Click ***Save***

LOCK method

METHOD DESCRIPTOR

Method	Level	Visibility	Description
LOCK	Static	Public	Write dataset lock entry

METHOD PARAMETERS

Click the *Parameters* button to define these

Parameter	Type	Pass Val	Method	Type	Description
DATASET	Importing	X	Type	ZPLUNK_TXT255	Splunk Dataset
RESULT	Exporting	X	Type	ZPLUNK_TXT255	Result String

METHOD CODE

Click ***Save***

Click ***Back*** (green arrow)

Double-click KEYPAIRT

From menu *Utilities* ⟶ *More Utilities* ⟶ *Upload/Download* ⟶ *Upload*
Import code from *meth_lock.txt* at the link

http://code.google.com/p/sapninja/source/browse/trunk/splunk/meth_lock.txt

Click ***Save***

UNLOCK method

METHOD DESCRIPTOR

Method	Level	Visibility	Description
UNLOCK	Static	Public	Remove dataset lock entry

METHOD PARAMETERS

Click the *Parameters* button to define these

Parameter	Type	Pass Val	Method	Type	Description
DATASET	Importing	X	Type	ZPLUNK_TXT255	Splunk Dataset
RESULT	Exporting	X	Type	ZPLUNK_TXT255	Result String

METHOD CODE

Click ***Save***

Click ***Back*** (green arrow)

Double-click KEYPAIRT

From menu *Utilities* → *More Utilities* → *Upload/Download* → *Upload*
Import code from *meth_unlock.txt* at the link

http://code.google.com/p/sapninja/source/browse/trunk/splunk/meth_unlock.txt

Click ***Save***

TIMESTAMP method

METHOD DESCRIPTOR

Method	Level	Visibility	Description
TIMESTAMP	Static	Public	Return timestamp in Splunk log format

METHOD PARAMETERS

Click the *Parameters* button to define these

Parameter	Type	Pass Val	Method	Type	Description
RAWDATE	Importing	X	Type	ZPLUNK_DATE	Raw date
RAWTIME	Importing	X	Type	ZPLUNK_TIME	Raw time
RESULT	Exporting	X	Type	ZPLUNK_TXT255	Formatted Timestamp

METHOD CODE

Click ***Save***

Click ***Back*** (green arrow)

Double-click KEYPAIRT

From menu *Utilities* → *More Utilities* → *Upload/Download* → *Upload*
Import code from *meth_timestamp.txt* at the link

http://code.google.com/p/sapninja/source/browse/trunk/splunk/meth_timestamp.txt

Click ***Save***

CCMS_MTE_TOUCH method

METHOD DESCRIPTOR

Method	Level	Visibility	Description
CCMS_MTE_TOUCH	Static	Public	Record MTE collection timestamp for this host

METHOD PARAMETERS

Click the *Parameters* button to define these

Parameter	Type	Pass Val	Method	Type	Description
MTE	Importing	X	Type	ZPLUNK_TXT255	MTE Class
RESULT	Exporting	X	Type	ZPLUNK_TXT255	Result String

METHOD CODE

Click ***Save***

Click ***Back*** (green arrow)

Double-click KEYPAIRT

From menu *Utilities* → *More Utilities* → *Upload/Download* → *Upload*
Import code from *meth_ccms_mte_touch.txt* at the link

http://code.google.com/p/sapninja/source/browse/trunk/splunk/meth_ccms_mte_touch.txt

Click ***Save***

CCMS_MTE_GETTIME method

METHOD DESCRIPTOR

Method	Level	Visibility	Description
CCMS_MTE_GETTIME	Static	Public	Retrieve MTE collection timestamp for this host

METHOD PARAMETERS

Click the *Parameters* button to define these

Parameter	Type	Pass Val	Method	Type	Description
MTE	Importing	X	Type	ZPLUNK_TXT255	MTE Class
RESULT	Exporting	X	Type	ZPLUNK_TXT255	Result String

METHOD CODE

Click ***Save***

Click ***Back*** (green arrow)

Double-click KEYPAIRT

From menu *Utilities* → *More Utilities* → *Upload/Download* → *Upload*
Import code from *meth_ccms_mte_gettime.txt* at the link

http://code.google.com/p/sapninja/source/browse/trunk/splunk/meth_ccms_mte_gettime.txt

Click ***Save***

CONV_SECS_SINCE_EPOCH method

METHOD DESCRIPTOR

Method	Level	Visibility	Description
CONV_SECS_SINCE_EPOCH	Static	Public	Convert seconds since epoch to timestamp

METHOD PARAMETERS

Click the *Parameters* button to define these

Parameter	Type	Pass Val	Method	Type	Description
SECS_SINCE_EPOCH	Changing	X	Type	TIMESTAMP	UTC Time Stamp in Short Form

METHOD CODE

Click ***Save***
Click ***Back*** (green arrow)
Double-click KEYPAIRT
From menu *Utilities* → *More Utilities* → *Upload/Download* → *Upload*
Import code from *meth_conv_secs_since_epoch.txt* at the link

http://code.google.com/p/sapninja/source/browse/trunk/splunk/meth_conv_secs_since_epoch.txt

Click ***Save***

ACTIVATE THE CLASS

Start transaction ***SE24***

Enter *Object type* = ***ZPLUNK_COMMON***

Click ***Activate*** or (CTRL+F3)

Function Modules

ZPLUNK_GET function module

Start transaction ***SE37***

Enter *Function Module* = ***ZPLUNK_GET***

Click ***Create***

Enter *Function Module* = ***ZPLUNK_GET***

Enter *Function Group* = ***ZPLUNK_FGRP***

Enter *Short Text* = ***RFC for Splunk record retrieval***

Function Module	ZPLUNK_GET
Function Group	ZPLUNK_FGRP
Short Text	RFC for Splunk record retrieval

Go to ***Source Code*** tab

From menu *Utilities* → *More Utilities* → *Upload/Download* → *Upload*

Import code from *fm_zplunk_get.txt* at the link

http://code.google.com/p/sapninja/source/browse/trunk/splunk/fm_zplunk_get.txt

From menu *Goto -> Text Elements -> Text Symbols*

Click *Yes* to save the function module code when prompted

Create the following symbol

Sym	Text
001	;

Click ***Save***

Click ***Back*** (green arrow)

ATTRIBUTES TAB

Choose or set the radio button *Processing Type* = ***Remote-Enabled Module***

IMPORT TAB

Parameter Name	Typing	Optional	Pass Value	Associated Type
DATASET	TYPE		X	ZPLUNK_TXT255

EXPORT TAB

Parameter Name	Typing	Pass Value	Associated Type
RESULT	TYPE	X	STRINGTAB

Click ***Save***

Click ***Activate***

Include Programs

ZPLUNK0001 include program

Start transaction ***SE38***

Click ***Create***

Title	ZPLUNK_CAPTURE - global variables
Type	Include Program
Status	Customer Production Program
Application	Basis

Click ***Enter***

Set *Package* = ***ZPLUNK***

Click ***Enter***

From menu *Utilities* → *More Utilities* → *Upload/Download* → *Upload*

Import code from inc_*zplunk0001.txt* at the link

http://code.google.com/p/sapninja/source/browse/trunk/splunk/inc_zplunk0001.txt

ZPLUNK0002 include program

Start transaction ***SE38***

Click ***Create***

Title	ZPLUNK_CAPTURE - time functions
Type	Include Program
Status	Customer Production Program
Application	Basis

Click ***Enter***

Set *Package* = ***ZPLUNK***

Click ***Enter***

From menu *Utilities* → *More Utilities* → *Upload/Download* → *Upload*

Import code from inc_*zplunk0002.txt* at the link

http://code.google.com/p/sapninja/source/browse/trunk/splunk/inc_zplunk0002.txt

ZPLUNK0003 include program

Start transaction ***SE38***

Click ***Create***

Title	ZPLUNK_CAPTURE - SM04 User List
Type	Include Program
Status	Customer Production Program
Application	Basis

Click ***Enter***

Set *Package* = ***ZPLUNK***

Click ***Enter***

From menu *Utilities* → *More Utilities* → *Upload/Download* → *Upload*

Import code from inc_*zplunk0003.txt* at the link

http://code.google.com/p/sapninja/source/browse/trunk/splunk/inc_zplunk0003.txt

ZPLUNK0004 include program

Start transaction ***SE38***

Click ***Create***

Title	ZPLUNK_CAPTURE - SM04 Memory Use
Type	Include Program
Status	Customer Production Program
Application	Basis

Click ***Enter***

Set *Package* = ***ZPLUNK***

Click ***Enter***

From menu *Utilities* ⟶ *More Utilities* ⟶ *Upload/Download* ⟶ *Upload*

Import code from inc*_zplunk0004.txt* at the link

http://code.google.com/p/sapninja/source/browse/trunk/splunk/inc_zplunk0004.txt

ZPLUNK0005 include program

Start transaction ***SE38***

Click ***Create***

Title	ZPLUNK_CAPTURE - ST03N save results
Type	Include Program
Status	Customer Production Program
Application	Basis

Click ***Enter***

Set *Package* = ***ZPLUNK***

Click ***Enter***

From menu *Utilities* → *More Utilities* → *Upload/Download* → *Upload*

Import code from inc_*zplunk0005.txt* at the link

http://code.google.com/p/sapninja/source/browse/trunk/splunk/inc_zplunk0005.txt

ZPLUNK0006 include program

Start transaction *SE38*

Click ***Create***

Title	ZPLUNK_CAPTURE - ST03N collect data
Type	Include Program
Status	Customer Production Program
Application	Basis

Click ***Enter***

Set *Package* = *ZPLUNK*

Click ***Enter***

From menu *Utilities* → *More Utilities* → *Upload/Download* → *Upload*
Import code from inc_*zplunk0006.txt* at the link

http://code.google.com/p/sapninja/source/browse/trunk/splunk/inc_zplunk0006.txt

ZPLUNK0007 include program

Start transaction ***SE38***

Click ***Create***

Title	ZPLUNK_CAPTURE - SM21 system log
Type	Include Program
Status	Customer Production Program
Application	Basis

Click ***Enter***

Set *Package* = ***ZPLUNK***

Click ***Enter***

From menu *Utilities* → *More Utilities* → *Upload/Download* → *Upload*
Import code from inc_*zplunk0007.txt* at the link

http://code.google.com/p/sapninja/source/browse/trunk/splunk/inc_zplunk0007.txt

ZPLUNK0008 include program

Start transaction ***SE38***

Click ***Create***

Title	ZPLUNK_CAPTURE - CCMS alerts
Type	Include Program
Status	Customer Production Program
Application	Basis

Click ***Enter***

Set *Package* = ***ZPLUNK***

Click ***Enter***

From menu *Utilities* → *More Utilities* → *Upload/Download* → *Upload*
Import code from inc*_zplunk0008.txt* at the link

http://code.google.com/p/sapninja/source/browse/trunk/splunk/inc_zplunk0008.txt

ZPLUNK0009 include program

Start transaction ***SE38***

Click ***Create***

Title	ZPLUNK_CAPTURE - SM50 workload
Type	Include Program
Status	Customer Production Program
Application	Basis

Click ***Enter***

Set *Package* = ***ZPLUNK***

Click ***Enter***

From menu *Utilities* → *More Utilities* → *Upload/Download* → *Upload*
Import code from inc_*zplunk0009.txt* at the link

http://code.google.com/p/sapninja/source/browse/trunk/splunk/inc_zplunk0009.txt

ZPLUNK0010 include program

Start transaction ***SE38***

Click ***Create***

Title	ZPLUNK_CAPTURE - SP01 print jobs
Type	Include Program
Status	Customer Production Program
Application	Basis

Click ***Enter***

Set *Package* = ***ZPLUNK***

Click ***Enter***

From menu *Utilities* → *More Utilities* → *Upload/Download* → *Upload*

Import code from inc*_zplunk0010.txt* at the link

http://code.google.com/p/sapninja/source/browse/trunk/splunk/inc_zplunk00010.txt

Main Programs

ZPLUNK_CAPTURE

Start transaction ***SE38***

Click ***Create***

Title	Splunk Collector Data Capture
Type	Executable Program
Status	Customer Production Program
Application	Basis

Click ***Enter***

Set *Package* = ***ZPLUNK***

Click ***Enter***

From menu *Utilities* → *More Utilities* → *Upload/Download* → *Upload*
Import code from prg_*zplunk_capture.txt* at the link

http://code.google.com/p/sapninja/source/browse/trunk/splunk/prg_zplunk_capture.txt

From menu *Goto* -> *Text Elements* -> *Text Symbols*
Click ***Yes*** to save the function module code when prompted
Create the following symbols

Sym	Text
431	Time Zone
432	Purge data by age
433	Data retention days
434	Purge extracted data
500	General Parameters
530	Log entry time
531	Last round hour
532	Last round 30 mins
533	Last round 15 mins
534	Last round 10 mins
535	Last round 5 mins
536	Last round minute
537	Actual time
540	ST03N Workload
541	SM21 System Log
543	CCMS Alerts
544	ST03N User Txns
545	ST03N Frontend

546	Print Jobs
550	Measurement period
551	Previous 1 minute
552	Previous 5 mins
553	Previous 10 mins
554	Previous 15 mins
555	Previous 30 mins
556	Previous hour
557	Previous 8 hours
558	Previous 12 hours
559	Previous 24 hours
560	SM04 User List
561	SM04 Memory List
562	CCMS Current
563	SM50 Work Processes
600	Time Range Data
630	Point-in-Time Data

Click ***Save***
Click ***Back*** (green arrow)
Activate the program

ZPLUNKSAPMSM21_700

Start transaction ***SE38***

Click ***Create***

Title	Customised SAPMSM21 for Basis Rel. 700
Type	Executable Program
Status	Customer Production Program
Application	Basis

Click ***Enter***

Set *Package* = ***ZPLUNK***
Click ***Enter***

From menu *Utilities* → *More Utilities* → *Upload/Download* → *Upload*
Import code from prg_*zplunksapmsm21_700.txt* at the link

http://code.google.com/p/sapninja/source/browse/trunk/splunk/prg_zplunksapmsm21_700.txt

Activate the program

Section 2 – Propagate Collector Framework

Create variants

The collector is very easy to configure with regards to what time span it collects for. Some measures are transitory point-in-time snapshots (like SM50 views), while others collect a time range of information like ST03N or SM21. The collector for ST03N uses standard SAP function modules, but it is still reasonably CPU intensive. If your system is heavily loaded and close to capacity, I recommend scheduling ST03N hourly instead of every 15 minutes as proposed here. Some metrics are system-wide, like CCMS alerts and print jobs and should therefore only be scheduled on the central instance. I suggest the following variants for most customers.

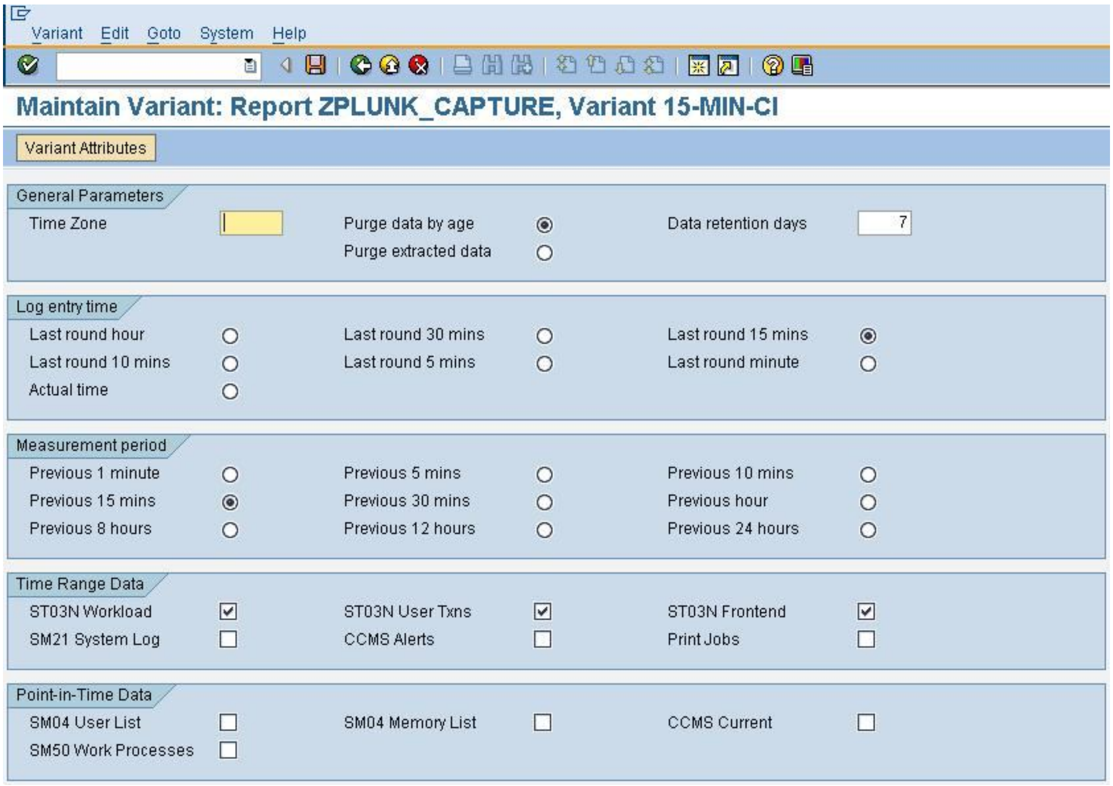
When you set up the variants, you choose whether the collector will delete data by age, or by whether it has been harvested by the *ZPLUNK_GET* function. I recommend doing it by age and keeping it for 7 days until you get used to the tools.

The variants below need to be created with transaction ***SE38*** with the ***variants*** radio button ticked and program ***ZPLUNK_CAPTURE*** selected.

LONG CYCLE ON CENTRAL INSTANCE

Variant	15-MIN-CI
Description	15 minute cycle for Central Instance

This variant will timestamp the data with the last round multiple of 15 minutes. For example,if it is 15:24, then the timestamp will be 15:15. The measurement span will be for 15 minutes, for the same example, then it would measure 15:00 to 15:15. Finally, only the ST03N monitors are selected.



Maintain Variant: Report ZPLUNK_CAPTURE, Variant 15-MIN-CI

General Parameters

Time Zone		Purge data by age	●	Data retention days	7
		Purge extracted data	○		

Log entry time

Last round hour	○	Last round 30 mins	○	Last round 15 mins	●
Last round 10 mins	○	Last round 5 mins	○	Last round minute	○
Actual time	○				

Measurement period

Previous 1 minute	○	Previous 5 mins	○	Previous 10 mins	○
Previous 15 mins	●	Previous 30 mins	○	Previous hour	○
Previous 8 hours	○	Previous 12 hours	○	Previous 24 hours	○

Time Range Data

ST03N Workload	☑	ST03N User Txns	☑	ST03N Frontend	☑
SM21 System Log	□	CCMS Alerts	□	Print Jobs	□

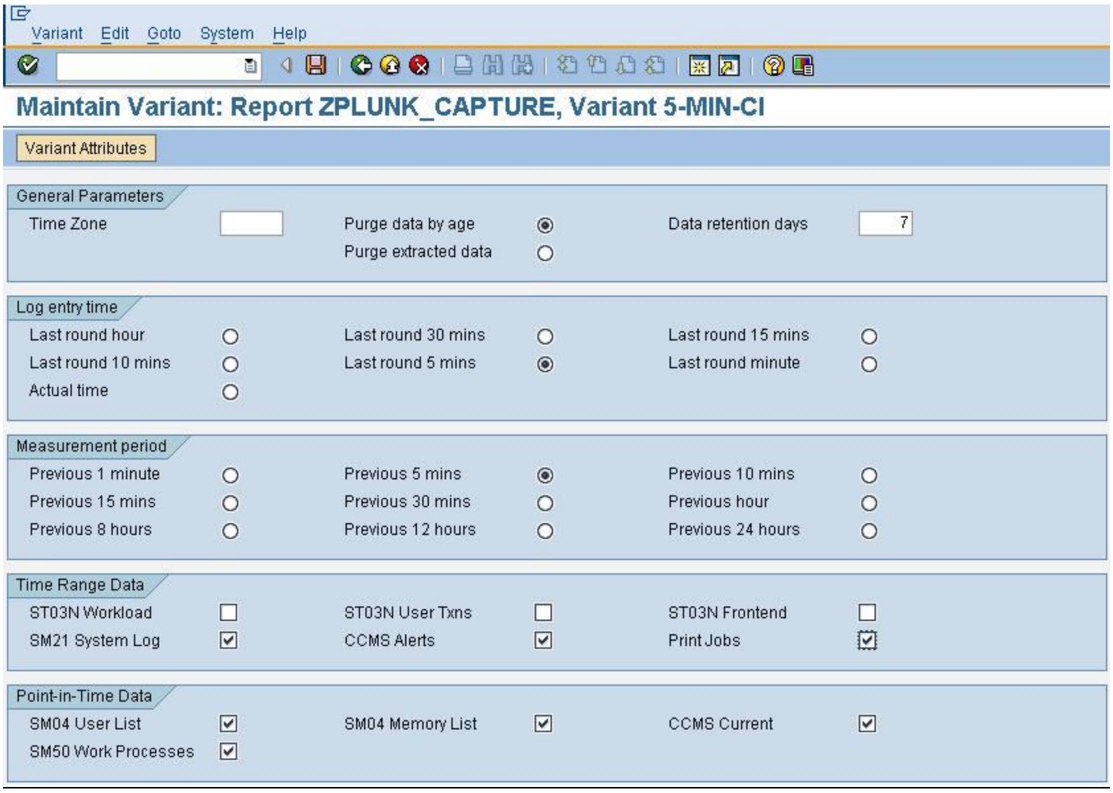
Point-in-Time Data

SM04 User List	□	SM04 Memory List	□	CCMS Current	□
SM50 Work Processes	□				

http://www.sapninja.com

SHORT CYCLE ON CENTRAL INSTANCE

Variant	5-MIN-CI
Description	5 minute cycle for Central Instance



Maintain Variant: Report ZPLUNK_CAPTURE, Variant 5-MIN-CI

General Parameters

Time Zone		Purge data by age	●	Data retention days	7
		Purge extracted data	○		

Log entry time

Last round hour	○	Last round 30 mins	○	Last round 15 mins	○
Last round 10 mins	○	Last round 5 mins	●	Last round minute	○
Actual time	○				

Measurement period

Previous 1 minute	○	Previous 5 mins	●	Previous 10 mins	○
Previous 15 mins	○	Previous 30 mins	○	Previous hour	○
Previous 8 hours	○	Previous 12 hours	○	Previous 24 hours	○

Time Range Data

ST03N Workload	□	ST03N User Txns	□	ST03N Frontend	□
SM21 System Log	☑	CCMS Alerts	☑	Print Jobs	☑

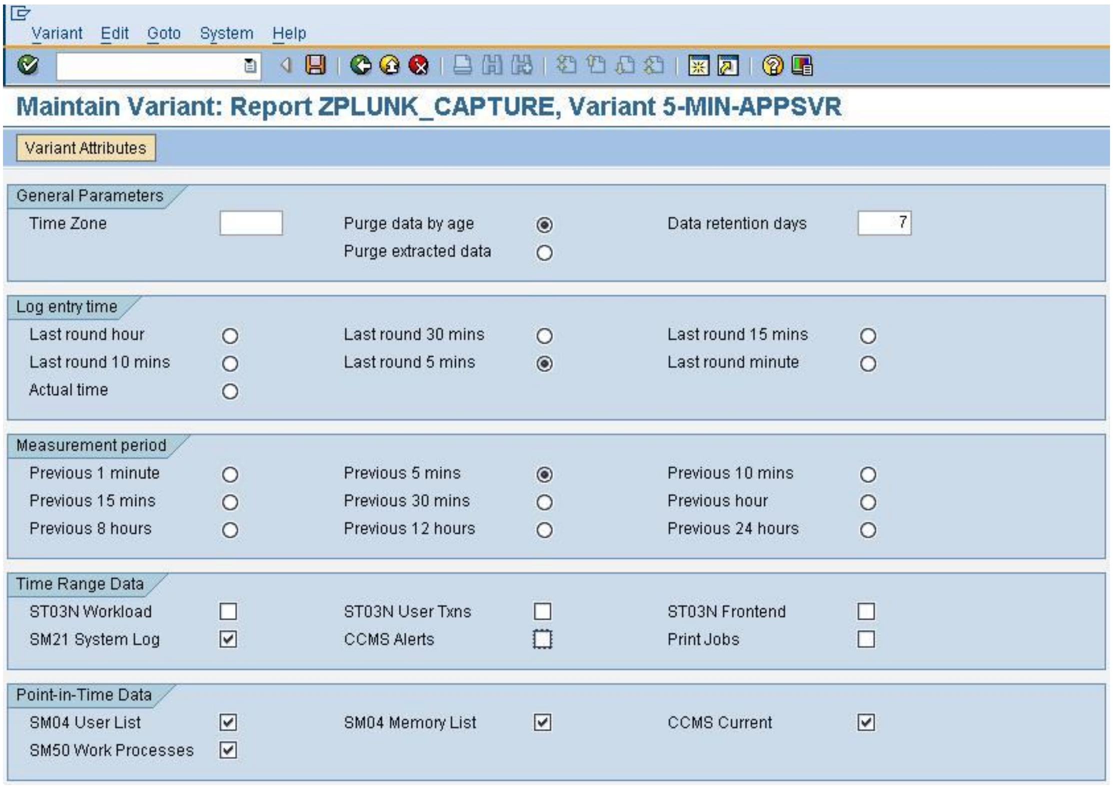
Point-in-Time Data

SM04 User List	☑	SM04 Memory List	☑	CCMS Current	☑
SM50 Work Processes	☑				

SHORT CYCLE ON APPLICATION SERVERS

The application server variant excludes CCMS Alerts and Print Jobs. Print jobs are system-wide so it makes sense to execute from the central instance variant. CCMS alerts are taken from the ALALERTS file, which is located centrally on the GLOBAL file system directory.

Variant	5-MIN-APPSVR
Description	5 minute cycle for Application Servers



Maintain Variant: Report ZPLUNK_CAPTURE, Variant 5-MIN-APPSVR

General Parameters

Time Zone		Purge data by age	●	Data retention days	7
		Purge extracted data	○		

Log entry time

Last round hour	○	Last round 30 mins	○	Last round 15 mins	○
Last round 10 mins	○	Last round 5 mins	●	Last round minute	○
Actual time	○				

Measurement period

Previous 1 minute	○	Previous 5 mins	●	Previous 10 mins	○
Previous 15 mins	○	Previous 30 mins	○	Previous hour	○
Previous 8 hours	○	Previous 12 hours	○	Previous 24 hours	○

Time Range Data

ST03N Workload	□	ST03N User Txns	□	ST03N Frontend	□
SM21 System Log	☑	CCMS Alerts	□	Print Jobs	□

Point-in-Time Data

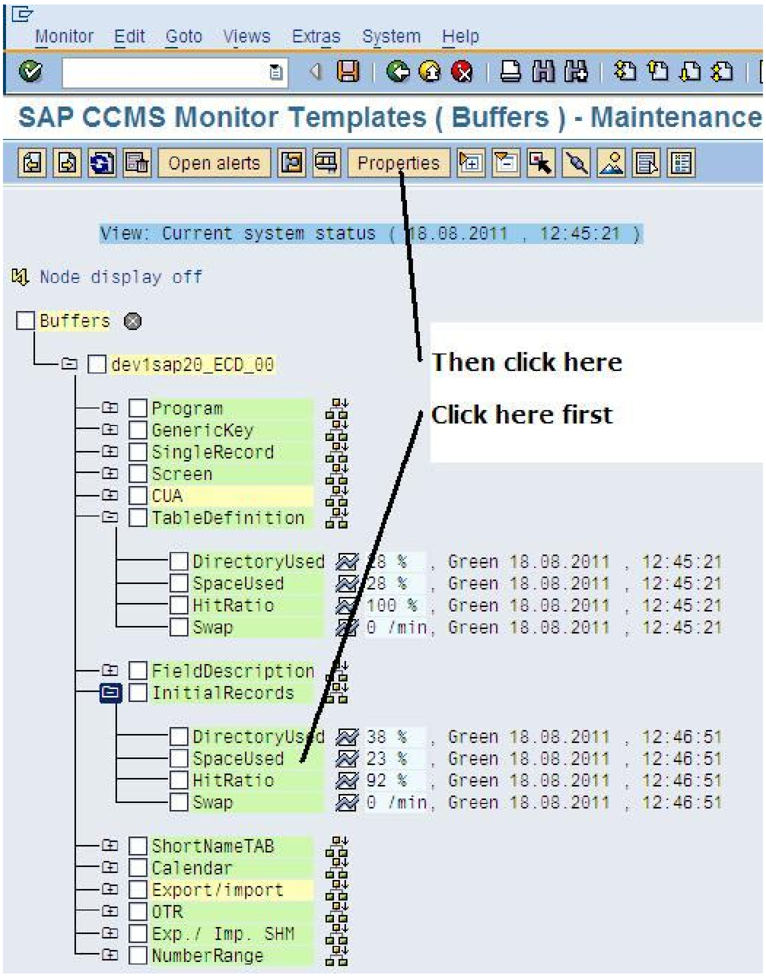
SM04 User List	☑	SM04 Memory List	☑	CCMS Current	☑
SM50 Work Processes	☑				

Configure ZPLUNK_CCMS_MTES

Give a man a fish, and you feed him for a day. Teach a man to fish and you feed him for life.

This section is not going to tell you what MTE's to monitor. It will only explain how to add MTE's to be collected with the "CCMS Current" option ticked.

By way of example, if you wanted to collect the CCMS for space used by the Initial Records buffer, this is what you would do. First, find the MTE in transaction RZ20.

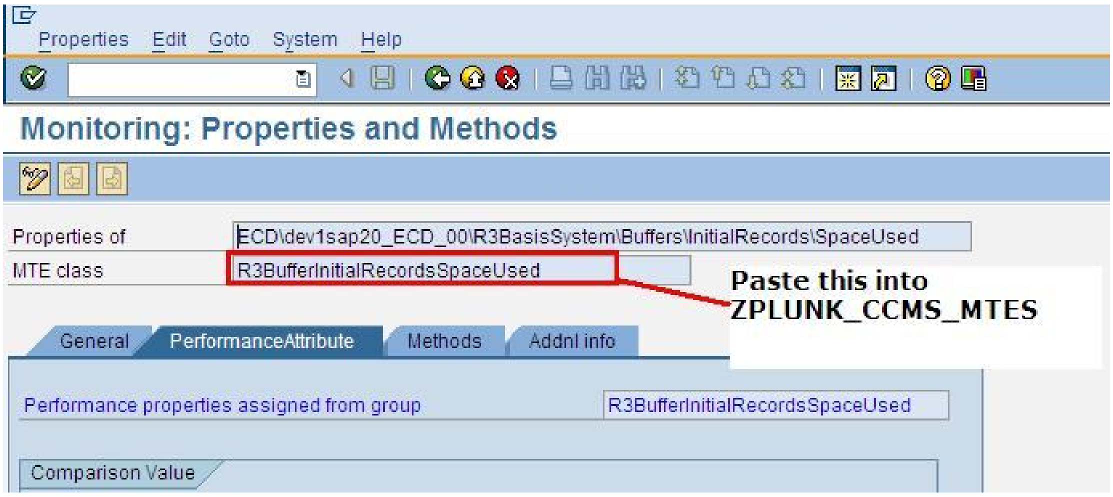


```
View: Current system status ( 18.08.2011 , 12:45:21 )

Node display off

□ Buffers
  □ dev1sap20_ECD_00
    □ Program
    □ GenericKey
    □ SingleRecord
    □ Screen
    □ CUA
    □ TableDefinition
      □ DirectoryUsed  18 %   , Green 18.08.2011 , 12:45:21
      □ SpaceUsed      28 %   , Green 18.08.2011 , 12:45:21
      □ HitRatio       100 %  , Green 18.08.2011 , 12:45:21
      □ Swap           0 /min, Green 18.08.2011 , 12:45:21
    □ FieldDescription
    □ InitialRecords
      □ DirectoryUsed  38 %   , Green 18.08.2011 , 12:46:51
      □ SpaceUsed      23 %   , Green 18.08.2011 , 12:46:51
      □ HitRatio       92 %   , Green 18.08.2011 , 12:46:51
      □ Swap           0 /min, Green 18.08.2011 , 12:46:51
    □ ShortNameTAB
    □ Calendar
    □ Export/import
    □ OTR
    □ Exp./ Imp. SHM
    □ NumberRange
```

Then click here

Click here first

This will bring you to the money screen. Grab hold of the highlighted value.



Monitoring: Properties and Methods

Properties of: ECD\dev1sap20_ECD_00\R3BasisSystem\Buffers\InitialRecords\SpaceUsed

MTE class: R3BufferInitialRecordsSpaceUsed

Paste this into ZPLUNK_CCMS_MTES

Performance properties assigned from group: R3BufferInitialRecordsSpaceUsed

Start transaction ***SM30*** and edit the table ***ZPLUNK_CCMS_MTES***. Paste in this MTE name as a new value. Repeat for every MTE that you want to capture a current value snapshot for.

Note It is not necessary to do this for alerts. All CCMS alerts will be fetched regardless of what is in this table

Create transport

As you have been following the steps in this document, you have probably already assigned these objects to a transport request which you can use to push the collector to downstream systems.

To create a new complete transport, with everything you have just set up:

Start transaction ***SE80***

Select the object type ***Package***

Enter ***ZPLUNK***

Click on the ***Glasses*** to display the package

Right-click on ***ZPLUNK*** and choose ***Write transport entry***. When prompted, choose ***All Objects***

To transport the variants, start transaction ***SA38*** and execute report ***RSTRANSP***. Enter ***ZPLUNK_CAPTURE*** as the program name and leave the variant name blank. Click execute and tick all the variants displayed. Assign them to your package transport.

To include the values that you coded into table ***ZPLUNK_CCMS_MTES***, execute transaction ***SE10***, find your transport, double-click on it to edit the objects and manually add and object entry

R3TR	TABU	ZPLUNK_CCMS_MTES

Enter a key value of "*" to select all keys. If this freaks you out a bit, you can edit that table manually in each of your downstream systems with transaction ***SM30***.

Schedule collectors

Now you have all the ingredients to schedule the collectors. I am not going to cover what security the user account needs, as I typically execute these jobs with a privileged system user. It is important to schedule a collector on each ABAP application server by specifying the execution host when creating the job from transaction ***SM36***.
I recommend scheduling the following jobs

ZPLUNK_CAPTURE_15MINS_<CENTRAL INSTANCE HOST>

Execution host = <Central Instance Host>
Report = ZPLUNK_CAPTURE
Variant = 15-MIN-CI
Frequency = 15 minutes
Start at = 00:03

ZPLUNK_CAPTURE_15MINS_<APPLICATION SERVER>

Execution host = <Each application server>
Report = ZPLUNK_CAPTURE
Variant = 15-MIN-CI (not a mistake)
Frequency = 15 minutes
Start at = 00:03

ZPLUNK_CAPTURE_5MINS_<CENTRAL INSTANCE HOST >

Execution host = <Central Instance Host>
Report = ZPLUNK_CAPTURE
Variant = 5-MIN-CI
Frequency = 5 minutes
Start at = 00:01

ZPLUNK_CAPTURE_5MINS_<APPLICATION SERVER>

Execution host = <Each application server>
Report = ZPLUNK_CAPTURE
Variant = 5-MIN-APPSVR
Frequency = 5 minutes
Start at = 00:01

Section 3 – Deploy Perl Extractor

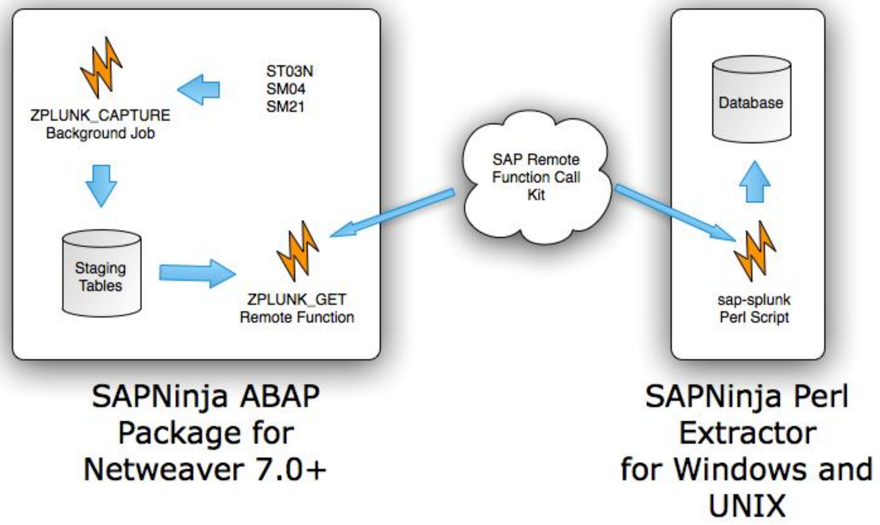
Once you get to this point on your journey of installation, you should have

- Installed all of the necessary SAP code
- Created job variants
- Set up users and security required
- Scheduled the collector jobs

In simple terms, you should have some ABAP systems which have harvested lots of useful data. Now you are ready to ingest this data into a tool like Splunk (or Excel, SQL Server or whatever analytical tools you use).

The best way to extract captured information remotely from SAP is to use Perl to make remote function calls to the SAP system. This requires the use of the SAP Netweaver RFC Software Development Kit (NWRFCSDK). Ordinarily, one would need C/Perl sofware development skills to achieve this, but Piers Harding is to be acknowledged and thanked for his contributions in making this reachable by members of the general community. This site will endeavour to simplify the instructions even further, so they become basic instructions that any Windows or LINUX administrator can follow. If you wish to explore this area further, please visit Piers' site at http://search.cpan.org/dist/sapnwrfc/sapnwrfc.pm.

I have supplied instructions of how to perform this for Linux or Windows systems. This will hopefully give you enough clues if you need to do it for a different operating system like AIX or HP-UX. Of course, if you have to do this, please share the instructions with sapsplunk@gmail.com and we will incorporate it into the documentation for the community.


ZPLUNK_CAPTURE Background Job
ST03N
SM04
SM21
Staging Tables
ZPLUNK_GET Remote Function
SAP Remote Function Call Kit
Database
sap-splunk Perl Script
SAPNinja ABAP Package for Netweaver 7.0+
SAPNinja Perl Extractor for Windows and UNIX

Security for the Extractors

Pre-prepared RFC Role

In this section, you will prepare the system to respond to external RFC calls by creating a user account with the appropriate access.

Use transaction **PFCG** in the SAP system to import the role ***ZSAPNINJA_RFC.SAP***. Choose ***Role → Upload*** from the menu. You can download this file from the link http://code.google.com/p/sapninja/source/browse/trunk/splunk/ZSAPNINJA_RFC.SAP

From SAP transaction ***SU01***, create a user (say ***SPLUNKRFC***) of type ***Communications user*** and assign it the role of ***ZSAPNINJA_RFC***

Security Technical Details

For data to be successfully extracted by the SAPninja perl script, you will need a user account in the target SAP system you are querying. The user only needs very minimal security. The account should be set up as a Communications User type so that the password never expires and nobody can log in interactively. The exact security required is as follows:

Auth Object	RFC Type	RFC Name	Activity
S_RFC	FUGR	SYST	16
S_RFC	FUGR	RFC1	16
S_RFC	FUGR	SDIFRUNTIME	16
S_RFC	FUGR	ZPLUNK_GET	16

NB: The security above is sufficient for the account to perform the extractions, but not enough to pass the perl module installation tests. This will not prevent your installation working successfully, but will throw up error messages that you probably don't want to see at the time. The following security should be temporarily enabled for the purposes of SAPNW::Rfc perl module installation and then immediately removed.

Auth Object	ACTVT	DICBERCLS	
S_TABU_DIS	03	SS	

Auth Object	DEVCLASS	OBJTYPE	OBJNAME	P_GROUP	ACTVT
S_DEVELOP	SRCX	FUGR	GRFC		03

Perl extractors on Linux

These instructions were developed on Red Hat Linux.

Install C Compiler

To install the module, you will need a gcc compiler on the system.

```
sudo yum install gcc
```

Acquire SAP NW RFC SDK libraries

You will need a valid SAP Marketplace account to download the nwrfcsdk libraries from SAP. At the time of writing, there were two versions available, 7.10 and 7.11. SAP indicate that the SDK libraries are backward-compatible, so it is a good idea to obtain the latest version. Be very sure to choose the version that matches your operating system architecture exactly.

The package is delivered as a *.SAR file, which is the SAP equivalent of tar in UNIX. To unpack it, you need the SAPCAR executable, which is also available from http://service.sap.com/patches. The navigation path will depend on your version but will be similar to this "Support Packages and Patches -> Entry by Application Group -> SAP NetWeaver and complementary products -> SAP NETWEAVER -> SAP NETWEAVER 7.0 -> Entry by Component" Application Server ABAP -> SAP KERNEL 7.00 64-BIT UNICODE -> Linux on x86_64 64bit -> #Database independent". You may need to rename the SAPCAR executable to remove any version information that forms part of its name, so that the file is called SAPCAR (or SAPCAR.exe if you use Windows).

Open http://service.sap.com/patches

Navigate to ***Entry by application group* → *Additional Components* → *SAP NWRFCSDK* → *SAP NW RFC SDK 7.10* → *Linux on x86_64 64bit*** (or whatever you are using)

Download the SAR file

Unpack the SAR file with SAPCAR and save as c:\nwrfcsdk. To unpack the archive, the command will look similar to this

```
SAPCAR -xvf NWRFC_8-20004549.SAR
```

Copy the unpacked directory nwrfcsdk to a permanent location, say, /usr/sap/nwrfcsdk

INSTALL ADDITIONAL PERL MODULES

You need to have ActiveState Perl installed before doing these steps. Install the Perl prerequisite packages:
Open a windows command prompt and type the commands below

```
cpan ExtUtils::MakeMaker
cpan YAML
```

Acquire CPAN module for perl and sap

All prerequisites are now in place for the compilation and installation of the SAPNW::Rfc perl module. Obtain the sapnwrfc-0.31.tar.gz gzipped tarball from CPAN (http://search.cpan.org/CPAN/authors/id/P/PI/PIERS/sapnwrfc-0.31.tar.gz).
Unpack the tarball

```
tar -xvzf sapnwrfc-0.31.tar.gz
```

Change directory to the unpacked directory.

```
cd sapnwrfc-0.31
```

Edit the file ***sap.yml*** and customize the following parameters to match your own installation. This is required for perl module compilations tests to work.

```
ashost: <your sap server host>
sysnr: "<your sap system number>"
client: "<your sap client>"
user: SPLUNKRFC
passwd: <password of SPLUNKRFC>
lcheck: 1
lang: EN
trace: 0
debug: 0
```

Compiling the extractor

It is now time to compile and install the SAPNW::Rfc perl module. This can be done with the following commands. Note: the user on the SAP target system will need to have slightly elevated privileges to allow the successful completion of the tests in the make test command below.

```
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/sap/nwrfcsdk/lib
echo "/usr/sap/nwrfcsdk/lib" > /etc/ld.so.conf.d/sapnwrfcsdk.conf
/sbin/ldconfig
perl Makefile.PL
make
# For the next command, ignore the failures from test 06 onwards.
make test
make install
```

Create YAML connection file

SAPninja Perl Script - Deploy the splunk-sap.pl perl script on your Splunk server. It should be placed where your other scripted input scripts are located. Create a yml connection file called sap.yml. The contents need to look like this, but with the correct values for host, user, password, etc.

```
ashost: <your sap server host>
sysnr: "<your sap system number>"
client: "<your sap client>"
user: SPLUNKRFC
passwd: <password of SPLUNKRFC>
lcheck: 1
lang: EN
trace: 0
debug: 0
```

The splunk-sap.pl script can be executed as follows. Ensure that your LD_LIBRARY_PATH variable has been populated with the location of the SAP Netweaver libraries (i.e. /usr/sap/nwrfcsdk/lib). Download the sample perl script splunk-sap.pl. You can download this file from the link http://code.google.com/p/sapninja/source/browse/trunk/splunk/splunk-sap.pl

```
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/sap/nwrfcsdk/lib
./splunk-sap.pl ./sap.yml ZPLUNK_SM50
```

Perl extractors on Windows

It is more difficult to set up the Perl extractors to Splunk on Windows than it is on LINUX due to peculiarities of the Windows C++ compilers and the environment generally. The instructions on this page will hopefully give you a sure-fire way to get one started before you try and use later versions of Visual C++ , Perl, or other components. I know for sure that these instructions will work with Visual C++ ExClick on Windows XP or Server 2003. If you manage to get the extractor working with later versions, we would appreciate you sharing the instructions so we can add them to this document.

Install C++

Install Visual C++ 2005 ExClick (Download from http://www.softpedia.com/progDownload/Microsoft-Visual-C-Toolkit-Download-11595.html)

Install Microsoft Platform SDK

Install MS Platform SDK for Windows XP/Server 2003 (Download from http://download.cnet.com/Windows-Server-2003-SP1-Platform-SDK-ISO-Install/3000-2070_4-10731571.html and rename from *.img to *.iso)

Install perl

Install ActiveState Perl (http://www.perl.org) . Navigate to the download page. Download and install ActivePerl version 5.10

Acquire SAP NW RFC SDK libraries

You will need a valid SAP Marketplace account to download the nwrfcsdk libraries from SAP. Proceed as follows

Navigate to http://service.sap.com/patches

Navigate to ***Entry by application group*** → ***Additional Components*** → ***SAP NW RFCSDK*** → ***SAP NW RFC SDK 7.10*** → ***Windows Server on IA32 32bit*** (or whatever you are using)

Download the SAR file

Unpack the SAR file with ***SAPCAR*** and save as c:\nwrfcsdk. If you cannot find ***SAPCAR***, please read the section that corresponds to this one within the section “Perl extractors on Linux”.

Acquire CPAN module for perl and sap

Download the CPAN Perl module for NWRFCSDK. Proceed as follows:

Navigate to http://search.cpan.org/dist/sapnwrfc/

Click the ***Download*** button to download sapnwrfc-0.31.tar.tar

Use ***winrar*** or similar to unpack the file to c:\sapnwrfc-0.31

INSTALL ADDITIONAL PERL MODULES

You need to have ActiveState Perl installed before doing these steps. Install the Perl prerequisite packages. Open a windows command prompt and type the commands below

```
cpan ExtUtils::MakeMaker
cpan YAML
```

Compiling the extractor

This activity can sort out the men from the boys. Proceed as follows:

Start → All Programs → Microsoft Platform SDK for Windows Server 2003 SP1

Choose ***Open build environment window***

Choose ***Windows XP 32bit Build Environment*** (or Server 2003 if that is what you are using)

Choose ***Set Windows XP 32bit Build Environment (Retail)*** (or the corresponding W2K3 server option)

This will open up a command prompt which has all sorts of compiler-centric settings ready on it. Set up your environment variables by typing in the command shown on the next bullet points

```
"C:\Program Files\Microsoft Visual Studio 8\VC\vcvarsall.bat" x86
cd C:\sapnwrfc-0.31
perl Makefile.PL --addlibs ' -lm -ldl -lpthread '
/*Enter [c:/nwrfcsdk] if prompted*/
```

There will be a file called ***Makefile*** in your directory which was created by the command above. Edit this file to get compiler and linker to use options from ***SAP Note 1056696***. You need to find the lines ***CCFLAGS*** and ***LDLOADLIBS*** and replace them with the following:

```
OSS_CCFLAGS = -DBCDASM -nologo -Od -Ob1 -fp:strict -Gy -GF -EHs -Z7 -W3 -Wp64 -
D_X86_ -DWIN32 -DSAPwithUNICODE -DUNICODE -D_UNICODE -MD -D_AFXDLL -FR -J
-RTC1 -D_CRT_NON_CONFORMING_SWPRINTFS -D_CRT_SECURE_NO_DEPRECATE
-D_CRT_NONSTDC_NO_DEPRECATE -DSAPonNT -c /EHc- /TP
CCFLAGS = $(CCFLAGS) $(OSS_CCFLAGS)
OSS_LDLOADLIBS = -nologo /NXCOMPAT -STACK:0x800000 ole32.lib rpcrt4.lib
oleaut32.lib oledb.lib uuid.lib kernel32.lib advapi32.lib user32.lib gdi32.lib winspool.lib
ws2_32.lib Iphlpapi.lib netapi32.lib comdlg32.lib shell32.lib dbghelp.lib version.lib mpr.lib
secur32.lib -OPT:REF -LARGEADDRESSAWARE -subsystem:console -out:*.exe *.obj
sapnwrfc.lib libsapucum.lib sapdecfICUlib.lib
LDLOADLIBS = $(LDLOADLIBS) $(OSS_LDLOADLIBS)
```

Compile the libraries by executing the command

```
nmake
```

Create YAML connection file

Edit the file sap.yml(which is a connection file) so it is configured for your system

```
ashost: <your sap server host>
sysnr: "<your sap system number>"
client: "<your sap client>"
user: SPLUNKRFC
passwd: <password of SPLUNKRFC>
lcheck: 1
lang: EN
trace: 0
debug: 0
```

Test your installation by executing

```
nmake test
```

This will have some failures, but none should be due to failed RFC connections or authentication problems.

Install your Perl module so it is ready for use by executing

```
nmake install
```

Download the sample perl script **splunk-sap.pl**. You can download this file from the link http://code.google.com/p/sapninja/source/browse/trunk/splunk/splunk-sap.pl . Modify the contents of this script to meet your needs, in particular which dataset you want to extract. A good one to start with is ZPLUNK_SM50.

Perform a test execution of your splunk extractor

Open a windows command prompt

```
cd /d c:\sapnwrfc-0.31
perl splunk-sap.pl sap.yml ZPLUNK_SM04USR
```

Note You need to have scheduled the ***ZPLUNK_CAPTURE*** data gathering jobs on the SAP system to collect information that can be returned by this call.

Note If you execute this command for a second time, it will not return any data that you have extracted already. You will never get the same information twice if you retrieve the data using the ***ZPLUNK_GET*** function module. If you are testing, you will need to re-execute ***ZPLUNK_CAPTUR***E on the server for the relevant data set to get some more data.

Note You can pipe the output of the command file to a text file by suffixing a "> file.txt" at the end of the example perl statement above.