Hewlett Packard
Enterprise

Flex Development

Flex Time

Brad Lee
May 2016

Contents

1.1 If logs have only Time info

Log sample

There is no YEAR, Month and Day info in the logs. Only current time is logged in the file.

```
[00:02:33-INFO] connection closed
```

FlexConnector

```
# FlexAgent Regex Configuration File
do.unparsed.events=true

regex=\\[(\\d{2}\:\\d{2}\:\\d{2})\\-(\\w{1,5})\\s?\\] (.*)

token.count=3

token[0].name=Timestamp
token[0].type=String

token[1].name=EventType
token[1].type=String

token[2].name=EventName
token[2].type=String


#submessage.messageid.token=
#submessage.token=


event.deviceVendor=__getVendor(Hello)
event.deviceProduct=__stringConstant(Hello)
event.deviceCustomString1=EventType
event.name=EventName
event.deviceReceiptTime=__createOptionalTimeStampFromString(__concatenate(__regexToken(__createTimeStampStri
ngFromSecondsMicros(__longToTimeStamp(__currentTimestampInSeconds()),000),"(\\d{1,2} \\S{3} \\d{4}).*"),"
",Timestamp),dd MMM yyyy HH\:mm\:ss)

#l10n.filename.prefix=
```

Result

Get the year, day info from the SmartConnector and concatenate with time info from the log

Manager Receipt Time ↑ 1	End Time ⇕	Name ⇕	Device Receipt Time	F
26 Apr 2016 14:25:08 PDT	26 Apr 2016 00:02:33 PDT	connection closed	26 Apr 2016 00:02:33 PDT	

1.2 If timestamp have no YEAR info

Sample log

```
Oct 22 11:23:33-INFO connection closed
```

Flexconnector

```
# FlexAgent Regex Configuration File
do.unparsed.events=true

regex=(\\w+ \\d+ \\d{2}\:\\d{2}\:\\d{2})\\-(\\w{1,5})\\s? (.*)

token.count=3

token[0].name=Timestamp
token[0].type=TimeStamp
token[0].format=MMM dd HH\:mm\:ss

token[1].name=EventType
token[1].type=String

token[2].name=EventName
token[2].type=String


#submessage.messageid.token=
#submessage.token=



event.deviceVendor=__getVendor(Hello)
event.deviceProduct=__stringConstant(Hello)
event.deviceCustomString1=EventType
event.name=EventName
event.deviceReceiptTime=__setYearToCurrentYear(Timestamp)


#l10n.filename.prefix=
```

Result

Today is 26 April 2016. But first event have 2015 time stamp with this parser. This is due to syslog.future.limit setting.

Manager Receipt Time ↑ 1	End Time ⇕	Name ⇕	Device Receipt Time
26 Apr 2016 15:10:23 PDT	22 Oct 2016 11:23:33 PDT	connection closed	22 Oct 2016 11:23:33 PDT
26 Apr 2016 15:05:05 PDT	22 Oct 2015 11:23:33 PDT	connection closed	22 Oct 2015 11:23:33 PDT

#agent.wrapper.conf

```
# if the difference between currentdate and apparentdate is more than these
# many days, we assume that it is from the past. set to -1 to disable.
```

After I set the “to syslog.future.limit=-1” in the agent.properties, I got the 2016 as a YEAR info.

1.3 If timestamp have mixed format

Sample log

```
1aug2015 21:02:36-INFO connection closed
```

Flexconnector

```
# FlexAgent Regex Configuration File
do.unparsed.events=true

regex=(\\d+\\w+\\d+) (\\d+\:\\d+\:\\d+)-(\\w{1,5})\\s? (.*)

token.count=4

token[0].name=_date
token[0].type=String

token[1].name=_time
token[1].type=String

token[2].name=EventType
token[2].type=String

token[3].name=EventName
token[3].type=String


#submessage.messageid.token=
#submessage.token=



event.deviceVendor=__getVendor(Hello)
event.deviceProduct=__stringConstant(Hello)
event.deviceCustomString1=EventType
event.name=EventName
event.deviceReceiptTime=__safeToDate(__concatenate(_date," ",_time),"dMMMyyyy HH\:mm\:ss")


#l10n.filename.prefix=
```

Result

Manager Receipt Time ↑ 1	End Time ⇕	Name ⇕	Device Receipt Time
26 Apr 2016 15:53:27 PDT	1 Aug 2015 21:02:36 PDT	connection closed	1 Aug 2015 21:02:36 PDT

1.4 If log filename have year,month,date info and log file have time info

Sample log

```
21:02:36-INFO connection closed
```

Log file name

```
aimlog20120625-1.txt
```

Flexconnector

```
# FlexAgent Regex Configuration File
do.unparsed.events=true

regex=(\\d+\:\\d+\:\\d+)-(\\w{1,5})\\s? (.*)

token.count=4

token[0].name=mytime
token[0].type=String

token[1].name=EventType
token[1].type=String

token[2].name=EventName
token[2].type=String

token[3].name=EventName
token[3].type=String


#submessage.messageid.token=
#submessage.token=



event.deviceVendor=__getVendor(Hello)
event.deviceProduct=__stringConstant(Hello)
event.deviceCustomString2=mytime
event.deviceCustomString1=EventType
event.name=EventName


#l10n.filename.prefix=
```

Map.0.properties file

```
set.expr(deviceCustomString2|deviceCustomString3).event.deviceCustomString4
"__concatenate(deviceCustomString3,"" "",deviceCustomString2)"
```

Agent.properties

```
agents[0].foldertable[0].extractfieldnames=deviceCustomString3
agents[0].foldertable[0].extractregex=aimlog(\\d+).*
agents[0].foldertable[0].extractsource=File Name
agents[0].foldertable[0].usefieldextractor=true
```

Result

Device Custom String2 have time, device Custom String3 have date info. And finally Device Custom String4 have Full timestamp.

Manager Receipt Time ↑ 1	End Time ⇕	Name ⇕	Device Custom String2	Device Custom String3	Device Custom String4
26 Apr 2016 16:08:53 PDT	26 Apr 2016 16:08:56 PDT	connection closed	21:02:36	20120625	20120625 21:02:36

1.5 Add 1 day to timestamp

Sample log

```
2015-04-12 21:02:36-INFO connection closed
```

Flexconnector

```
# FlexAgent Regex Configuration File
do.unparsed.events=true

regex=(\\d+-\\d+-\\d+) (\\d+\:\\d+\:\\d+)-(\\w{1,5})\\s? (.*)

token.count=4

token[0].name=mydate
token[0].type=String

token[1].name=mytime
token[1].type=String

token[2].name=EventType
token[2].type=String

token[3].name=EventName
token[3].type=String


#submessage.messageid.token=
#submessage.token=

event.deviceVendor=__getVendor(Hello)
event.deviceProduct=__stringConstant(Hello)
event.deviceCustomString1=EventType
event.name=EventName
event.deviceReceiptTime=__createLocalTimeStampFromSecondsSinceEpoch(__integerToLong(__sum(__divide(__toLong
TimeStamp(__concatenate(mydate," ",mytime)), 1000), 86400)))

#l10n.filename.prefix=
```

Result

On the Device Receipt Time, the date is added by 1 day

Manager Receipt Time ↑ 1	End Time	Name	Device Receipt Time
26 Apr 2016 17:30:00 PDT	13 Apr 2015 21:02:36 PDT	connection closed	13 Apr 2015 21:02:36 PDT

1.6 Timestamp have time and elapsed time

Sample log

```
11OCT2011 00:00:00 5:55:55-INFO connection closed
```

Flexconnector

```
# FlexAgent Regex Configuration File
do.unparsed.events=true

regex=(\\d+\\w+\\d+ \\d+\:\\d+\:\\d+) (\\d+\:\\d+\:\\d+)-(\\w{1,5})\\s? (.*)

token.count=4

token[0].name=starttime
token[0].type=String

token[1].name=elapsedtime
token[1].type=String

token[2].name=EventType
token[2].type=String

token[3].name=EventName
token[3].type=String


#submessage.messageid.token=
#submessage.token=



event.deviceVendor=__getVendor(Hello)
event.deviceProduct=__stringConstant(Hello)
event.deviceCustomString1=EventType
event.name=EventName
event.deviceReceiptTime=__createTimeStampByStartTimeElapsed(starttime,elapsedtime)


#l10n.filename.prefix=
```

Result

Manager Receipt Time ↑ 1	End Time	Name	Device Receipt Time
26 Apr 2016 18:06:18 PDT	11 Oct 2011 05:55:55 PDT	connection closed	11 Oct 2011 05:55:55 PDT

1.7 Create local time from GMT

Sample log

```
May 08 2010 14:24:5-INFO connection closed
```

Flexconnector

```
# FlexAgent Regex Configuration File
do.unparsed.events=true

regex=(\\w+ \\d+ \\d+) (\\d+\:\\d+\:\\d+)-(\\w{1,5})\\s? (.*)

token.count=4

token[0].name=mydate
token[0].type=Date
token[0].format=MMM dd yyyy

token[1].name=mytime
token[1].type=Time
token[1].format=HH\:mm\:s

token[2].name=EventType
token[2].type=String

token[3].name=EventName
token[3].type=String


#submessage.messageid.token=
#submessage.token=



event.deviceVendor=__getVendor(Hello)
event.deviceProduct=__stringConstant(Hello)
event.deviceCustomString1=EventType
event.name=EventName
event.deviceReceiptTime=__createLocalTimeStampFromGMT(mydate,mytime)

#l10n.filename.prefix=
```

Result

Manager Receipt Time ↑ 1	End Time ⇕	Name ⇕	Device Receipt Time
26 Apr 2016 18:13:41 PDT	8 May 2010 07:24:05 PDT	connection closed	8 May 2010 07:24:05 PDT

1.8 Replace time zone

Example

Replaces the Z with a UTC and assigns it to the deviceReceiptTime.

```
conditionalmap.count=1
conditionalmap[0].field=event.deviceVendor
conditionalmap[0].mappings.count=2
conditionalmap[0].mappings[0].values=FireEye
conditionalmap[0].mappings[0].event.deviceReceiptTime=__parseMutableTimeStamp(__replaceAll(rt,Z,UTC))
conditionalmap[0].mappings[1].event.deviceReceiptTime=__parseMutableTimeStamp(rt)
```

1.9 Second, microsecond timestamp and conditional mapping

Sample log

```
1 0.0-INFO connection closed
1 1456500428.493521492-INFO connection closed
```

Flexconnector

```
# FlexAgent Regex Configuration File
do.unparsed.events=true

regex=\\d (\\d+).(\\d+)-(\\w{1,5})\\s? (.*)

token.count=4

token[0].name=second
token[0].type=Long

token[1].name=microsecond
token[1].type=Long

token[2].name=EventType
token[2].type=String

token[3].name=EventName
token[3].type=String


#submessage.messageid.token=
#submessage.token=



event.deviceCustomNumber3=second
event.deviceVendor=__getVendor(Hello)
event.deviceProduct=__stringConstant(Hello)
event.deviceCustomString1=EventType
event.name=EventName

event.deviceReceiptTime=__createLocalTimeStampFromSecondsMicrosZone(second,microsecond,PST)

conditionalmap.count=1
conditionalmap[0].field=event.deviceCustomNumber3
conditionalmap[0].mappings.count=1
conditionalmap[0].mappings[0].values=0
conditionalmap[0].mappings[0].event.deviceReceiptTime=__createOptionalTimeStampFromString(__regexToken(__crea
teTimeStampStringFromSecondsMicros(__currentTimestampInSeconds(),0),"(.*)"),dd MMM yyyy HH:mm:ss)

#l10n.filename.prefix=
```

Result

If the timestamp(second,microsecond) is valid, map it to DeviceReceiptTime. But if the time is '0', get the local timestamp on the SmartConnector server and map it to DeviceReceiptTime

Manager Receipt Time 1	End Time	Name	Device Receipt Time	Device Custom Number3
26 Apr 2016 18:29:11 PDT	26 Apr 2016 18:29:19 PDT	connection closed	26 Apr 2016 18:29:19 PDT	0
26 Apr 2016 18:29:11 PDT	26 Feb 2016 07:35:21 PST	connection closed	26 Feb 2016 07:35:21 PST	1456500428

1.10 Timestamp have starttime and endtime

Sample log

It have starttime and endtime

```
1426657844342.1426657907617-INFO connection closed
```

Flexconnector

```
# FlexAgent Regex Configuration File
do.unparsed.events=true

regex=(\\d+).(\\d+)-(\\w{1,5})\\s? (.*)

token.count=4

token[0].name=starttime
token[0].type=String

token[1].name=endtime
token[1].type=String

token[2].name=EventType
token[2].type=String

token[3].name=EventName
token[3].type=String


#submessage.messageid.token=
#submessage.token=


event.deviceCustomDate1=__longToTimeStamp(__safeToLong(starttime))
event.deviceCustomDate2=__longToTimeStamp(__safeToLong(endtime))
event.deviceVendor=__getVendor(Hello)
event.deviceProduct=__stringConstant(Hello)
event.deviceCustomString1=EventType
event.name=EventName

#l10n.filename.prefix=
```

Result

Manager Receipt Time ↑ 1	Device Custom Date1	Device Custom Date2
26 Apr 2016 18:47:36 PDT	17 Mar 2015 22:50:44 PDT	17 Mar 2015 22:51:47 PDT

1.11 Create time from epochtime

Sample log

```
1353995148.439-INFO connection closed
```

Flexconnector

```
# FlexAgent Regex Configuration File
do.unparsed.events=true

regex=(\\d+).\\d+-(\\w{1,5})\\s? (.*)

token.count=4

token[0].name=epochtime
token[0].type=Long

token[1].name=EventType
token[1].type=String

token[2].name=EventName
token[2].type=String

token[3].name=EventName
token[3].type=String


#submessage.messageid.token=
#submessage.token=



event.deviceVendor=__getVendor(Hello)
event.deviceProduct=__stringConstant(Hello)
event.deviceCustomString1=EventType
event.name=EventName
event.deviceReceiptTime=__createLocalTimeStampFromSecondsSinceEpoch(epochtime)

#l10n.filename.prefix=
```

Result

Manager Receipt Time ↑ 1	End Time ⇕	Device Receipt Time
26 Apr 2016 19:01:20 PDT	26 Nov 2012 21:45:48 PST	26 Nov 2012 21:45:48 PST

1.12 Create time from epochtime and millisecond

Sample log

```
1353995148.439-INFO connection closed
```

Flexconnector

```
# FlexAgent Regex Configuration File
do.unparsed.events=true

regex=(\\d+).(\\d+)-(\\w{1,5})\\s? (.*)

token.count=4

token[0].name=epochtime
token[0].type=Long

token[1].name=millisecond
token[1].type=Long

token[2].name=EventType
token[2].type=String

token[3].name=EventName
token[3].type=String


#submessage.messageid.token=
#submessage.token=

event.deviceVendor=__getVendor(Hello)
event.deviceProduct=__stringConstant(Hello)
event.deviceCustomString1=EventType
event.name=EventName
event.deviceReceiptTime=__createLocalTimeStampFromGMTSecondsMillis(epochtime,millisecond)


#l10n.filename.prefix=
```


Result

Manager Receipt Time 1	End Time	Device Receipt Time	Name
26 Apr 2016 19:09:45 PDT	26 Nov 2012 13:45:48 PST	26 Nov 2012 13:45:48 PST	connection closed

If only use the epochtime.
event.deviceReceiptTime=__createLocalTimeStampFromSecondsSinceEpoch(epochtime)

26 Apr 2016 19:07:25 PDT	26 Nov 2012 21:45:48 PST	26 Nov 2012 21:45:48 PST	connection closed
26 Apr 2016 19:01:20 PDT	26 Nov 2012 21:45:48 PST	26 Nov 2012 21:45:48 PST	connection closed

1.13 Create time from epochtime and response time

Sample log

```
1166088812.985 199-INFO connection closed
```

Flexconnector

```
# FlexAgent Regex Configuration File
do.unparsed.events=true

regex=(\\d+).\\d+ (\\d+)-(\\w{1,5})\\s? (.*)

token.count=4

token[0].name=epochtime
token[0].type=String

token[1].name=ResponseTime
token[1].type=String

token[2].name=EventType
token[2].type=String

token[3].name=EventName
token[3].type=String


#submessage.messageid.token=
#submessage.token=


event.deviceCustomNumber1=__safeToLong(ResponseTime)
event.deviceVendor=__getVendor(Hello)
event.deviceProduct=__stringConstant(Hello)
event.deviceCustomString1=EventType
event.deviceCustomNumber1Label=__stringConstant("Response Time")
event.name=EventName
event.deviceReceiptTime=__createLocalTimeStampFromSecondsSinceEpoch(__safeToLong(epochtime))


#l10n.filename.prefix=
```

Result

Manager Receipt Time ↑ 1	End Time ⇕	Device Receipt Time	Name ⇕	Device Cu
26 Apr 2016 19:19:42 PDT	14 Dec 2006 01:33:32 PST	14 Dec 2006 01:33:32 PST	connection closed	199

1.14 Various time stamp

Sample log

```
2/01/11 1:38 AM-INFO connection closed
2.02.11 10:47 PM-INFO connection closed
2.02.11 11:57-INFO connection closed
7-02-11 15:16-INFO connection closed
```

Flexconnector

```
# FlexAgent Regex Configuration File
do.unparsed.events=true

regex=(\\d+[/|.|-]\\d+[/|.|-]\\d+) (\\d+\:\\d+\\s?\\w+?)-(\\w{1,5})\\s?(.*)

token.count=4

token[0].name=Date
token[0].type=String

token[1].name=Time
token[1].type=String

token[2].name=EventType
token[2].type=String

token[3].name=EventName
token[3].type=String


#submessage.messageid.token=
#submessage.token=



event.deviceVendor=__getVendor(Hello)
event.deviceProduct=__stringConstant(Hello)
event.deviceCustomString1=EventType
event.name=EventName
event.deviceReceiptTime=__oneOfDateTime(__safeToDate(__concatenate(Date," ",Time),"dd/MM/yy K\:mm
a"),__safeToDate(__concatenate(Date," ",Time),"d.MM.yy HH\:mm a"),__safeToDate(__concatenate(Date,"
",Time),"dd.MM.yy HH\:mm"),__safeToDate(__concatenate(Date," ",Time),"dd-MM-yy HH\:mm"))


#l10n.filename.prefix=
```


Result

Manager Receipt Time 1	End Time	Name	Device Receipt Time
27 Apr 2016 10:06:07 PDT	2 Jan 2011 01:38:00 PST	connection closed	2 Jan 2011 01:38:00 PST
27 Apr 2016 10:06:07 PDT	2 Feb 2011 10:47:00 PST	connection closed	2 Feb 2011 10:47:00 PST
27 Apr 2016 10:06:07 PDT	2 Feb 2011 11:57:00 PST	connection closed	2 Feb 2011 11:57:00 PST
27 Apr 2016 10:06:07 PDT	7 Feb 2011 15:16:00 PST	connection closed	7 Feb 2011 15:16:00 PST

1.15 Timestamp with offset

Sample log

```
2016-03-14 15:37:06-0400 [-] OVPN 10 OUT: 'Mon Mar 14 19:37:06 2016 166.171.250.165:60398
```

Flexconnector

```
# FlexAgent Regex Configuration File
do.unparsed.events=true

regex=(.*? \\d+\:\\d+\:\\d+)(-\\d{4}) [^ ]+ (\\w+) \\d+ (\\w+)\: '(.*) (\\d{0,3}.\\d{0,3}.\\d{0,3}.\\d{0,3})\:(\\d{0,5})

token.count=7

token[0].name=syslogTime
token[0].type=TimeStamp
token[0].format=yyyy-MM-dd HH\:mm\:ss

token[1].name=offset
token[1].type=String

token[2].name=eventName
token[2].type=String

token[3].name=deviceCustomStrings1
token[3].type=String

token[4].name=realtime
token[4].type=TimeStamp
token[4].format=E MMM dd HH\:mm\:ss

token[5].name=sourceAddress
token[5].type=IPAddress

token[6].name=sourcePort
token[6].type=Integer


#submessage.messageid.token=
#submessage.token=



event.deviceCustomDate1=syslogTime
event.sourcePort=sourcePort
event.deviceCustomString1=deviceCustomStrings1
event.name=eventName
event.deviceTimeZone=__getTimeZone(offset)
event.sourceAddress=sourceAddress
event.deviceReceiptTime=realtime
```

#l10n.filename.prefix=

Result

This is PDT time. And Device Time Zone is Anguilla

Manager Receipt Time ↑ 1	End Time	Name	Device Receipt Time	Device Time Zone	
27 Apr 2016 14:46:15 PDT	14 Mar 2016 19:37:06 PDT	OVPN	14 Mar 2016 19:37:06 PDT	America/Anguilla	

1.16 HexEncoded Time stamp

Sample log

```
23041909181C-INFO connection closed
```

Flexconnector

```
# FlexAgent Regex Configuration File
do.unparsed.events=true

regex=([^-]+)-(\\w{1,5})\\s? (.*)

token.count=3

token[0].name=HexEncodedTime
token[0].type=String

token[1].name=deviceCustomString
token[1].type=String

token[2].name=eventName
token[2].type=String


#submessage.messageid.token=
#submessage.token=



event.deviceVendor=__getVendor(Hello)
event.deviceProduct=__stringConstant(Hello)
event.deviceCustomString1=deviceCustomString
event.name=eventName
event.deviceReceiptTime=__createTimeStampByHexEncodedTime(HexEncodedTime)


#l10n.filename.prefix=
```

Result

Manager Receipt Time 1	End Time	Device Receipt Time	Name	A
2 May 2016 13:53:44 PDT	5/25 9:24:28	25 May 2005 09:24:28 PDT	connection closed	

1.17 Multiple Time Stamp

Sample log

```
1 2012-02-07T11:04:56.901 fake.dbdbd.ie RT_FLOW
1 2013-09-26T08:11:03.735+02:00 GBF-FW-MGT RT_FLOW
1 2012-02-07T11:04:56.901 dfw03.FAKE2.ie RT_FLOW
1 2012-04-03T17:39:50.750-0200 MADEUPTIMESTAMPFORTEST RT_FLOW
1 2012-04-03T17:39:50+0200 MADEUPTIMESTAMPFORTEST RT_FLOW
1 2012-04-03T17:39:50.750Z MADEUPTIMESTAMPFORTEST RT_FLOW
1 2012-04-03T17:39:50Z MADEUPTIMESTAMPFORTEST RT_FLOW
1 2012-04-03T17:39:50 MADEUPTIMESTAMPFORTEST RT_FLOW
1 2012-04-19T14:59:58.307 nyoFAKEyl-f1 RT_FLOW
```

Flexconnector

```
# FlexAgent Regex Configuration File
do.unparsed.events=true

regex=\\d+ (\\d{4}-\\d{2}-\\d{2}T\\d{2}\\:\\d{2}\\:\\d{2}(?\:\\.\\d{1,3})?(?\:Z|[+-]?\\d{4})?(?\:Z|[+-
]?\\d{2}\\:\\d{2})?)\\s(\\S+) (.*)

token.count=3

token[0].name=datetime
token[0].type=String

token[1].name=deviceCustomString
token[1].type=String

token[2].name=eventName
token[2].type=String


#submessage.messageid.token=
#submessage.token=



event.deviceVendor=__getVendor(Hello)
event.deviceProduct=__stringConstant(Hello)
event.deviceCustomString1=deviceCustomString
event.name=eventName
event.deviceReceiptTime=__parseMultipleTimeStamp(datetime,"yyyy-MM-dd'T'HH\:mm\:ss.SSSZ","yyyy-MM-
dd'T'HH\:mm\:ssZ","yyyy-MM-dd'T'HH\:mm\:ss.SSS","yyyy-MM-dd'T'HH\:mm\:ss")


#l10n.filename.prefix=
```


Result

Manager Receipt Time 1	Name	Device Receipt Time	Device Custom String1
2 May 2016 15:44:53 PDT	RT_FLOW	7 Feb 2012 11:04:56 PST	fake.dbdbd.ie
2 May 2016 15:44:53 PDT	RT_FLOW	26 Sep 2013 08:11:03 PDT	GBF-FW-MGT
2 May 2016 15:44:53 PDT	RT_FLOW	7 Feb 2012 11:04:56 PST	dfw03.FAKE2.ie
2 May 2016 15:44:53 PDT	RT_FLOW	3 Apr 2012 12:39:50 PDT	MADEUPTIMESTAMPFORTEST
2 May 2016 15:44:53 PDT	RT_FLOW	3 Apr 2012 08:39:50 PDT	MADEUPTIMESTAMPFORTEST
2 May 2016 15:44:53 PDT	RT_FLOW	3 Apr 2012 17:39:50 PDT	MADEUPTIMESTAMPFORTEST
2 May 2016 15:44:53 PDT	RT_FLOW	3 Apr 2012 17:39:50 PDT	MADEUPTIMESTAMPFORTEST
2 May 2016 15:44:53 PDT	RT_FLOW	3 Apr 2012 17:39:50 PDT	MADEUPTIMESTAMPFORTEST
2 May 2016 15:44:53 PDT	RT_FLOW	19 Apr 2012 14:59:58 PDT	nyoFAKEyl-f1

Timestamp in the Log	Device Receipt Time from the ESM
2012-02-07T11:04:56.901	7 Feb 2012 11:04:56 PST
2013-09-26T08:11:03.735+02:00	26 Sep 2013 08:11:03 PDT
2012-02-07T11:04:56.901	7 Feb 2012 11:04:56 PST
2012-04-03T17:39:50.750-0200	3 Apr 2012 12:39:50 PDT
2012-04-03T17:39:50+0200	3 Apr 2012 08:39:50 PDT
2012-04-03T17:39:50.750Z	3 Apr 2012 17:39:50 PDT
2012-04-03T17:39:50Z	3 Apr 2012 17:39:50 PDT
2012-04-03T17:39:50	3 Apr 2012 17:39:50 PDT
2012-04-19T14:59:58.307	19 Apr 2012 14:59:58 PDT

#Time when the message was generated,in one of two representations:

YYYY-MM-DDTHH:MM:SS.MSZ is the year, month, day, hour, minute,second and millisecond in Universal Coordinated Time (UTC)

YYYY-MM-DDTHH:MM:SS.MS+/-HH:MM is the year, month, day, hour, minute, second and millisecond in local time;

the hour and minute that follows the plus sign (+) or minus sign (-) is the offset of the local time zone from UTC

1.18 Unicode date convert – month.. Jan.. Feb..

Sample log

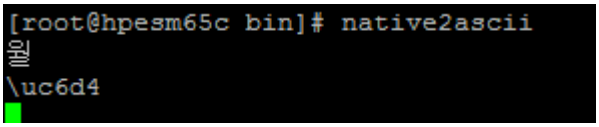
Chinese character. Each month presented as Chinese character.

```
11 一月 2014 10:14:52 CST Event_1
11 二月 2014 10:14:52 CST Event_2
```

How to convert to Unicode

Use the below link to convert or native2ascii command

http://www.branah.com/unicode-converter



Flexconnector

```
regex=.*?(\\d+) (.*) (\\d+) (\\d\\d\:\\d\\d\:\\d\\d) CST(.*)
token.count=5
token[0].name=date
token[0].type=String
token[1].name=month
token[1].type=String
token[2].name=year
token[2].type=String
token[3].name=hour
token[3].type=String
token[4].name=name
token[4].type=String
event.deviceVendor=__stringConstant("TEST")
event.name=name
event.deviceCustomString1=month
event.deviceProduct=__stringConstant("TimeFormat")

conditionalmap.count=1
conditionalmap[0].field=event.deviceCustomString1
conditionalmap[0].mappings.count=2
conditionalmap[0].mappings[0].values=\u4E8C\u6708
conditionalmap[0].mappings[0].event.deviceCustomString2=__concatenate(date,"Feb",year," ",hour)
conditionalmap[0].mappings[0].event.endTime=__createTimeStampForOpsecStartTime(__concatenate
(date,"Feb",year," ",hour))
conditionalmap[0].mappings[1].values=\u4E00\u6708
```

```
conditionalmap[0].mappings[1].event.deviceCustomString2=__concatenate(date,"Jan",year," ",hour)
conditionalmap[0].mappings[1].event.endTime=__createTimeStampForOpsecStartTime(__concatenate
(date,"Jan",year," ",hour))
```

Result

Manager Receipt Time ↑ 1	End Time	Name	Device Vendor	Device Product
11 May 2016 18:56:29 KST	11 Jan 2014 10:14:52 KST	Event_1	TEST	TimeFormat
11 May 2016 18:56:29 KST	11 Feb 2014 10:14:52 KST	Event_2	TEST	TimeFormat

1.19 Unicode date convert – month.

Sample log

Korean character – Month is represented in Korean Character. Replace the "월" to whitespace. 11 is Nov, so no need to process local character in the parser.

```
10.220.64.66 - - [09/11 월/2014:16:19:09 +0900] "POST /ktrerp/xp/executeds HTTP/1.1" 200 843
10.220.64.66 - - [09/11 월/2014:16:19:09 +0900] "POST /ktrerp/xp/executeds HTTP/1.1" 200 512
10.220.64.66 - - [09/11 월/2014:16:19:09 +0900] "GET /ktrerp/xpforms/Main/main_item_b_no.xfdl HTTP/1.1" 304 0
10.220.64.66 - - [09/11 월/2014:16:19:09 +0900] "POST /ktrerp/xp/executeds HTTP/1.1" 200 633
10.220.64.66 - - [09/11 월/2014:16:19:10 +0900] "POST /ktrerp/xp/executeds HTTP/1.1" 200 661
```

Flexconnector

```
# FlexAgent Regex Configuration File

#10.10.10.10 - - [30/Nov/2011:20:57:44 -0800] "GET /index.jsp HTTP/1.1" 200 10628
#10.10.10.10 - - [08/Dec/2011:12:59:08 -0800] "/index.html.ol<" 400 0
#10.22.64.66 - - [09/11 월/2014:16:20:03 +0900] "POST /ktrerp/xp/executeds HTTP/1.1" 200 857
regex=(\\S+) (.*?) (.*?) \\[(.*\uC6D4.*)\\] "((?\:(.*?) )?(.*?)(?\: (.*?))?)" (\\d+) (\\d+)(?\:\\s*)
token.count=10

token[0].name=hostName
token[0].type=String

token[1].name=RFC931
token[1].type=String

token[2].name=authUser
token[2].type=String

token[3].name=date
token[3].type=String

token[4].name=request
token[4].type=String

token[5].name=method
token[5].type=String

token[6].name=requestURI
token[6].type=String

token[7].name=httpVersion
token[7].type=String

token[8].name=status
token[8].type=String
```



```
token[9].name=bytes
token[9].type=String


#submessage.messageid.token=
#submessage.token=


event.bytesOut=__safeToInteger(bytes)
event.sourceUserName=RFC931
event.deviceAction=status
event.name=__concatenate("Method: ",method," Error Code: ",status)
event.requestUrl=requestURI
event.destinationUserId=authUser
event.deviceSeverity=status
event.applicationProtocol=httpVersion
event.deviceEventClassId=status
event.requestMethod=method
#event.deviceReceiptTime=date
event.sourceHostName=hostName
event.deviceVendor=__stringConstant("Hello")
event.deviceProduct=__stringConstant("Hello")

event.endTime=__createOptionalTimeStampFromString(__replaceAll(date,\\uC6D4,),dd/MM/yyyy\:hh\:mm\:ss
Z)
```

Result

Manager Receipt Time ↑ 1	End Time	Name	Attacker Address	Device Vendor	Device Product
11 May 2016 19:01:21 KST	9 Nov 2014 16:19:09 KST	Method: POST Error Code: 200	10.220.64.66	Hello	Hello
11 May 2016 19:01:21 KST	9 Nov 2014 16:19:09 KST	Method: POST Error Code: 200	10.220.64.66	Hello	Hello
11 May 2016 19:01:21 KST	9 Nov 2014 16:19:09 KST	Method: GET Error Code: 304	10.220.64.66	Hello	Hello
11 May 2016 19:01:21 KST	9 Nov 2014 16:19:09 KST	Method: POST Error Code: 200	10.220.64.66	Hello	Hello
11 May 2016 19:01:21 KST	9 Nov 2014 16:19:10 KST	Method: POST Error Code: 200	10.220.64.66	Hello	Hello

1.20 Unicode date convert – AM/PM.

Sample log

AM/PM represented by local character.

```
2013/10/3 上午 07:58:43  Test
2013/10/3 下午 07:58:44  Test
```

```
2013/10/3 오전 07:58:43  Test
2013/10/3 오후 07:58:44  Test
```

Flexconnector

- 上午/下午

```
regex=(\\d+\\/\\d+\\/\\d+ (\u4E0A\u5348|\u4E0B\u5348) \\d\\d\:\\d\\d\:\\d\\d)(.*)
token.count=3
token[0].name=time
token[0].type=String
token[1].name=noon
token[1].type=String
token[2].name=msg
token[2].type=String
event.endTime=__createOptionalTimeStampFromString(__replaceAll(__replaceAll(time,\u4E0A\u5348,AM),\u4E0B\u5348,PM),yyyy/MM/dd
aa HH\:mm\:ss)
event.message=__replaceAll(__replaceAll(time,\u4E0A\u5348,am),\u4E0B\u5348,pm)
```

- 오전/오후

```
# FlexAgent Regex Configuration File
do.unparsed.events=true

regex=(\\d+\\/\\d+\\/\\d+ (\\uc624\\uc804|\\uc624\\ud6c4) \\d\\d\:\\d\\d\:\\d\\d)(.*)

token.count=3

token[0].name=time
token[0].type=String

token[1].name=noon
token[1].type=String

token[2].name=msg
```

```
token[2].type=String


#submessage.messageid.token=
#submessage.token=


event.endTime=__createOptionalTimeStampFromString(__replaceAll(__replaceAll(time,\uC624\uC804,AM),\uC624\uD6C4,PM),yyyy/MM
/dd aa hh\:mm\:ss)
event.message=__replaceAll(__replaceAll(time,\uC624\uC804,am),\uC624\uD6C4,pm)


#l10n.filename.prefix=
```

Manager Receipt Time ⇕ ↑ 1	End Time ⇕	Message
11 May 2016 18:47:18 KST	3 Oct 2013 07:58:43 KST	2013/10/3 am 07:58:43
11 May 2016 18:47:18 KST	3 Oct 2013 19:58:44 KST	2013/10/3 pm 07:58:44

Result

1.21 Unicode date convert, only have time info and minus 1 day

Sample log

This timestamp only have date info. No year, month and day info.

```
19:59:10-INFO connection closed
```

Flexconnector

```
# FlexAgent Regex Configuration File
do.unparsed.events=true

regex=(\\d{2}\:\\d{2}\:\\d{2})\\-(\\w{1,5})\\s?\\ (.*)

token.count=3

token[0].name=time
token[0].type=String

token[1].name=EventType
token[1].type=String

token[2].name=EventName
token[2].type=String


#submessage.messageid.token=
#submessage.token=


event.deviceVendor=__getVendor(Hello)
event.deviceProduct=__stringConstant(Hello)
event.deviceCustomString1=EventType
event.name=EventName
event.deviceReceiptTime=__createOptionalTimeStampFromString(__concatenate(__regexToken(__createTimeStampStri
ngFromSecondsMicros(__integerToLong(__subtract(__currentTimestampInSeconds(),86400)),0),"(\\d+\\s\\d\\S\\s\\d+).*
"),"  ",time),"dd M'\uC6D4' yyyy HH:mm:ss")
#l10n.filename.prefix=


#for English OS

#event.deviceReceiptTime=__createOptionalTimeStampFromString(__concatenate(__regexToken(__createTimeStampSt
ringFromSecondsMicros(__integerToLong(__subtract(__currentTimestampInSeconds(),86400)),0),"(\\d{1,2} \\S{3}
\\d{4}).*"),"  ",time),"dd MMM yyyy HH:mm:ss")
```

Result

Manager Receipt Time 1	End Time	Name	Device Vendor
5/13 9:10:46	5/12 19:59:10	connection closed	Hello

Parser detail

Unlike English Windows OS, local Windows OS have different timestamp format

[Example timestamp from SmartConnector OS]

```
"10 8 월 2014 11:11:10 KST"
```

- Subtract : to minus 1 days from current day

```
event.deviceReceiptTime=__createOptionalTimeStampFromString(__concatenate(__regexToken(__createTimeStampStri
ngFromSecondsMicros(__integerToLong(__subtract(__currentTimestampInSeconds(),86400)),0),"(\\d+\\s\\d\\S\\s\\d+).
*")," ",time),"dd M'\uC6D4' yyyy HH:mm:ss")
```

- regexToken : to extract the Year, Month and Day. And don't take the time part from the Windows OS time.

1. Original timestamp from SmartConnector OS

10 8 월 2014 11:11:10 KST

2. Parsed time stamp with regexToken

10 8 월 2014 11:11:10 KST

__regexToken(__createTimeStampStringFromSecondsMicros(__integerToLong(__subtract(__currentTimestampInSecon
ds(),86400)),0),"(\\d+\\s\\d\\S\\s\\d+).*")

3. Timestamp

- From OS : 10 8 월 2014 11:11:10 KST
- Remove local character: 10 8 2014 19:59:10. '\uC6D4' is Unicode character of 월. Enclose with ' char to remove it.
- 10 9 2014 : use regex to take timestamp from the OS
- time : 19:59:10 : take it from sample log.

__createOptionalTimeStampFromString(__concatenate(__regexToken(__createTimeStampStringFromSecondsMicros(__i
ntegerToLong(__subtract(__currentTimestampInSeconds(),86400)),0),"(\\d+\\s\\d\\S\\s\\d+).*")," ",time),"dd M'\uC6D4'
yyyy HH:mm:ss")


```
event.deviceReceiptTime=__createOptionalTimeStampFromString(__concatenate(__regexToken(__createTimeStampStr
ingFromSecondsMicros(__integerToLong(__subtract(__currentTimestampInSeconds(),86400)),0),"(\\d+\\s\\d\\S\\s\\d+).
*")," ",time),"dd M'\uC6D4' yyyy HH:mm:ss")
```

- createOptionalTimeStampFromString : make the timestamp from the String. Plus **\uC6D4** is the Unicode of "월"

"10 8 월 2014 19:59:10 KST"

```
event.deviceReceiptTime=__createOptionalTimeStampFromString(__concatenate(__regexToken(__createTimeStampS
tringFromSecondsMicros(__integerToLong(__subtract(__currentTimestampInSeconds(),86400)),0),"(\\d+\\s\\d\\S\\s\\d+
).*")," ",time),"dd M'\uC6D4' yyyy HH:mm:ss")
```