



ATS

- [Support for Advanced Request Sending Options, on page 1](#)
- [Support for Conditional LFS Message Stubbing, on page 2](#)
- [Support for Regular Expressions, on page 3](#)

Support for Advanced Request Sending Options

Feature Summary and Revision History

Table 1: Summary Data

Applicable Product(s) or Functional Area	ATS
Applicable Platform(s)	Not Applicable
Default Setting	Enabled - Always-on
Related Changes in This Release	Not Applicable
Related Documentation	Contact your Cisco Account representative

Table 2: Revision History

Revision Details	Release
First introduced	19.3.0

Feature Description

ATS now supports advanced request sending options as follows:

- REST sends synchronous and asynchronous requests without using a message reference.
- Utilizing dynamic rule values in URI of the REST request: The support is provided to use the Rules on Request URI of any load profile transactions. To use the rule value, the following value can be provided in the load testing REST send grammars.

{RULE.Rulename}

**Restriction**

Currently, REST requests are supported.

Deprecated grammars do not support this feature. This feature is supported only for new grammar with the transaction.

For more information, contact your Cisco Account representative.

Support for Conditional LFS Message Stubbing

Feature Summary and Revision History

Table 3: Summary Data

Applicable Product(s) or Functional Area	ATS
Applicable Platform(s)	Not Applicable
Default Setting	Enabled – Configuration Required
Related Changes in This Release	Not Applicable
Related Documentation	Contact your Cisco Account representative

Table 4: Revision History

Revision Details	Release
First introduced	19.3.0

Feature Description

ATS now provides support for conditional LFS Message Stubbing.

Details of the implementation on how specific steps from a feature file can be stubbed based on the criteria provided in the message bundle and config properties is captured in the *CPS Automation Testing Grammar Reference*.

The feature provides the following advantages:

- Specific steps of a complex call flow can be skipped easily with a change in the configuration and message bundle.
- If a transaction is skipped then its corresponding validation is skipped automatically.
- Skipping of step is also applicable where duplicate transaction ID is used.
- This feature also works for external feature file.

**Restriction**

Currently, this feature is available for LFS driver.

The skipping of the steps is currently only based on two parameter: InterfaceType and NodeName.

Skip is not supported if dynamic message filter is provided in the asynchronous call.

Configuration

To enable this feature, following properties needs to be added in the config properties file:

- **InterfaceType:** `<LFSInstanceName>.InterfaceType.Skip=<Comma separated interface names>`

Example:

```
LFS.Instance.Names=LFSInstance01
LFSInstance01.InterfaceType.Skip=N1, N2
```

- **NodeName:** `<LFSInstanceName>.NodeName.Skip=<Comma separated node names>`

Example:

```
LFS.Instance.Names=LFSInstance01
LFSInstance01.NodeName.Skip=amf1, amf2
```

For more information, contact your Cisco Account representative.

Support for Regular Expressions

Feature Summary and Revision History

Table 5: Summary Data

Applicable Product(s) or Functional Area	ATS
Applicable Platform(s)	Not Applicable
Default Setting	Enabled - Always-on
Related Changes in This Release	Not Applicable
Related Documentation	Contact your Cisco Account representative

Table 6: Revision History

Revision Details	Release
First introduced	19.3.0

Feature Description

ATS now provides support for the usage of regular expressions in the array when an LFS message contains multiple similar pattern entries. The following operations can be performed:

- Filtering pattern-based array element using the regular expression.
- Predicate now supports multiple conditions.
- Both '&&' and '||' is now supported in the predicate filtering.

The '&&' operator can be applied when multiple conditions are to be satisfied while filtering the predicate.

The '||' operator can be applied when any of the multiple conditions are to be satisfied while filtering the predicate.



Note While using the '||' operator in the feature file it needs to be added with an escape character such as, \\|



Restriction Parenthesis are not supported within the predicate to group multiple conditions.

Multiple conditions are sequential.

For more information, contact your Cisco Account representative.