



## Cisco XML and Large Data Retrieval (Iterators)

The extensible markup language (XML) for the router supports the retrieval of large XML responses in blocks (that is, in chunks or sections).

This chapter provides information on large data retrieval. For information on terminating an iterator, see [“Terminating an Iterator” section on page 7-92](#).

When a client application makes a request, the resulting response data size is checked to determine if it is larger than a predetermined block size. If it is not larger, then the complete data is returned in a normal response. However, if the response data is larger than the block size, then the first set of data is returned according to the block size along with an iterator ID included as the value of the `IteratorID` attribute. The client must then send `<GetNext>` requests including the iterator ID until all the data is retrieved. The client application knows that all of the data is retrieved when it receives a response that does not contain an `IteratorID` attribute.

The following points should be noted by the client application when iterators are used:

- The block size is a fixed value specific to each transport mechanisms on the router, that is, the XML agent for Common Object Request Broker Architecture (CORBA) and Secure Shell (SSH) or Telnet. No mechanism is provided for the client application to specify a desired block size.
- The block size refers to the entire XML response, not just the payload portion of the response.
- Large responses are divided based on the requested block size, not on the contents. However, each response is always a complete XML document.
- Requests containing multiple operations are treated as a single entity when the block size and `IteratorID` are applied. As a result, the `IteratorID` is an attribute of the `<Response>` tag, never of an individual operation.
- If the client application sends a request that includes an operation resulting in the need for an iterator to return all of the response data, any further operations contained within that request are rejected. The rejected operations are resent in another request.
- The `IteratorID` is an unsigned 32-bit value that should be treated as opaque data by the client application. Furthermore, the client application should not assume that the `IteratorID` is constant between `<GetNext>` operations.

To reduce memory overhead and avoid memory starvation of the router, the following limitations are placed on the number of allowed iterators:

- 10—Maximum number of iterators allowed at any one time on a given client session.
- 100—Maximum number of iterators allowed at any one time for all client sessions.

**Note**

If a <Get> request is issued that results in an iterated response, it is counted as 1 iterator, regardless of the number of <GetNext> operations required to retrieve all of the response data. For example, a <Get> request may require 10, 100, or more <GetNext> operations to retrieve all of the associated data, but during this process only 1 iterator is being used. Also, an iterator is considered to be in use until all of the response data associated with that iterator (that is, all of the response data associated with the original <Get> request) is retrieved or the iterator is terminated with the Abort attribute.

The following example shows a client request that utilizes an iterator to retrieve all global Border Gateway Protocol (BGP) configuration data for a specified autonomous system:

**Sample XML Client Request to Retrieve All BGP Configuration Data**

```
<?xml version="1.0" encoding="UTF-8"?>
<Request MajorVersion="1" MinorVersion="0">
  <Get>
    <Configuration>
      <BGP MajorVersion="1" MinorVersion="0">
        <AS>
          <Naming>
            <AS>3</AS>
          </Naming>
          <Global/>
        </AS>
      </BGP>
    </Configuration>
  </Get>
</Request>
```

**Sample XML Response from the Router Containing the First Block of Retrieved Data**

```
<?xml version="1.0" encoding="UTF-8"?>
<Response MajorVersion="1" MinorVersion="0" IteratorID="1">
  <Get>
    <Configuration>
      <BGP MajorVersion="1" MinorVersion="0">
        <AS>
          <Naming>
            <AS>3</AS>
          </Naming>
          <Global>
            .
            .
            .
            1st block of data returned here
            .
            .
            .
          </Global>
        </AS>
      </BGP>
    </Configuration>
  </Get>
</Response>
```

**Second XML Client Request Using the <GetNext> Iterator to Retrieve the Next Block of BGP Configuration Data**

```
<?xml version="1.0" encoding="UTF-8"?>
<Request MajorVersion="1" MinorVersion="0">
  <GetNext IteratorID="1"/>
</Request>
```

**Sample XML Response from the Router Containing the Second Block of Retrieved Data**

```
<?xml version="1.0" encoding="UTF-8"?>
<Response MajorVersion="1" MinorVersion="0" IteratorID="1">
  <Get>
    <Configuration>
      <BGP MajorVersion="1" MinorVersion="0">
        <AS>
          <Naming>
            <AS>3</AS>
          </Naming>
          <Global>
            .
            .
            .
            2nd block of data returned here
            .
            .
          </Global>
        </AS>
      </BGP>
    </Configuration>
  </Get>
</Response>
```

**Third XML Client Request Using the <GetNext> Iterator to Retrieve the Next Block of BGP Configuration Data**

```
<?xml version="1.0" encoding="UTF-8"?>
<Request MajorVersion="1" MinorVersion="0">
  <GetNext IteratorID="1"/>
</Request>
```

**Sample XML Response from the Router Containing Third Block of Retrieved Data**

```
<?xml version="1.0" encoding="UTF-8"?>
<Response MajorVersion="1" MinorVersion="0" IteratorID="1">
  <Get>
    <Configuration>
      <BGP MajorVersion="1" MinorVersion="0">
        <AS>
          <Naming>
            <AS>3</AS>
          </Naming>
          <Global>
            .
            .
            .
            3rd block of data returned here
            .
            .
          </Global>
        </AS>
      </BGP>
    </Configuration>
  </Get>
</Response>
```

**Final XML Client Request Using the <GetNext> Iterator to Retrieve the Last Block of BGP Configuration Data**

```
<?xml version="1.0" encoding="UTF-8"?>
<Request MajorVersion="1" MinorVersion="0">
  <GetNext IteratorID="1"/>
</Request>
```

**Final XML Response from the Router Containing the Final Block of Retrieved Data**

```

<?xml version="1.0" encoding="UTF-8"?>
<Response MajorVersion="1" MinorVersion="0">
  <Get>
    <Configuration>
      <BGP MajorVersion="1" MinorVersion="0">
        <AS>
          <Naming>
            <AS>3</AS>
          </Naming>
          <Global>
            .
            .
            .
            Final block of data returned here
            .
            .
          </Global>
        </AS>
      </BGP>
    </Configuration>
  </Get>
</Response>

```

## Terminating an Iterator

A client application may terminate an iterator without retrieving all of the response data by including an Abort attribute with a value of “true” on the <GetNext> operation. A client application that does not complete or terminate its requests risks running out of iterators.

The following example shows a client request using the Abort attribute to terminate an iterator:

**Sample XML Request**

```

<?xml version="1.0" encoding="UTF-8"?>
<Request MajorVersion="1" MinorVersion="0">
  <Get>
    <Configuration>
      <BGP MajorVersion="1" MinorVersion="0">
        <AS>
          <Naming>
            <AS>3</AS>
          </Naming>
          <Global/>
        </AS>
      </BGP>
    </Configuration>
  </Get>
</Request>

```

**Sample XML Response from the Router**

```

<?xml version="1.0" encoding="UTF-8"?>
<Response MajorVersion="1" MinorVersion="0" IteratorID="2">
  <Get>
    <Configuration>
      <BGP MajorVersion="1" MinorVersion="0">
        <AS>
          <Naming>
            <AS>3</AS>
          </Naming>
          <Global>
            .
            .
            .
            1st block of data returned here
            .
            .
            .
          </Global>
        </AS>
      </BGP>
    </Configuration>
  </Get>
</Response>

```

**Sample XML Request Using the Abort Attribute to Terminate an Iterator**

```

<?xml version="1.0" encoding="UTF-8"?>
<Request MajorVersion="1" MinorVersion="0">
  <GetNext IteratorID="2" Abort="true"/>
</Request>

```

**Sample XML Response from the Router**

```

<?xml version="1.0" encoding="UTF-8"?>
<Response MajorVersion="1" MinorVersion="0">
  <GetNext IteratorID="2" Abort="true"/>
</Response>

```

