

# New Features Bulletin Adaptive Server® 15.7 ESD #1

Document ID: DC00650-01-1570-01

Last revised: February 7, 2012

Topic	Page
Proxy table support for in-row LOB columns	1
New replicated system procedures	3
Increased number of parameters	3
Multiple listeners on Windows	4

## Proxy table support for in-row LOB columns

Use Adaptive Server® version 15.7 ESD #1 to create in-row large object (LOB) columns. However, because proxy tables do not store information about remote in-row LOB columns in their metadata, when information about the LOB column on the source or target table is unavailable, Adaptive Server stores data off-row on the target table.

These commands support proxy tables with in-row LOB data:

Command	Supported when:
<code>alter database ... for proxy_update</code>	The underlying proxy tables contain in-row LOB columns.
<code>alter table add   modify .... in row, ...</code>	These commands are forwarded directly to the remote server.
<code>create database...with default location...for proxy update</code>	The underlying proxy tables contain in-row LOB columns.
<code>create existing table .. (<i>column_name</i> in row, . . . ) at <i>external_location</i></code>	

Command	Supported when:
<code>create proxy_table ... at external_location</code>	The remote table contains in-row LOB columns.
<code>create table .. (column_name in row, ... ) at external_location</code>	The remote table contains in-row LOB columns; any proxy columns created do not contain in-row properties.
<code>insert into {local   proxy} table select {local   proxy} table</code>	The target data is stored in-row, if the size of the data is within the in-row limit.
<code>select into proxy_table from local_table</code>	The target data is stored in-row, if the size of the data is within the in-row limit.
<code>select into existing table proxy_table from local_table</code>	The target data is stored in-row, if the size of the data is within the in-row limit.
<code>select into proxy_table (column_name in row, ... ) from local_table</code>	The target data is stored in-row, if the size of the data is within the in-row limit.
<code>select into local_table (column_name in row ... ) from proxy_table</code>	The target data is stored in-row.
<code>update proxy_table set lob_column =</code>	The target data is stored in-row, if the size of the data is within the in-row limit.

These commands do not support proxy tables with in-row LOB data:

- `create database proxy_database ... inrow_lob_length = n`
- `alter database proxy_database ... set inrow_lob_length = n`

During these select into and bulk copy (bcp) operations, target data is stored in-row if the size of the data is within the in-row limit, and the source data is in-row (otherwise the target data is stored off-row):

- `select into local_table (column_name in row (n) ... ) from proxy_table` (target data will be stored in-row)
- `select into proxy_table from local_table`

- `select into proxy_table (column_name in row (n)... ) from local_table`
- `select into existing_table proxy_table from local_table`
- `select into local_table from proxy_table`
- `select into proxy_table_1 from proxy_table_2`
- `bcp database_name ... proxy_table in bcp_file ...`

## New replicated system procedures

Adaptive Server 15.7 ESD #1 allows replication for these system procedures in a replicated master database:

- `sp_addexternlogin`
- `sp_dropexternlogin`
- `sp_maplogin`
- `sp_addremotelogin`
- `sp_dropremotelogin`
- `sp_addserver`
- `sp_dropserver`

## New and changed configuration parameters

Adaptive Server version 15.7 ESD 1 adds the network polling mode configuration parameter.

### In-line network polling

Adaptive Server 15.7 spawns separate threads to perform network polling. However, separate threads may not show significant performance gains when running with a low engine configuration. Adaptive Server may suffer from contention between engines and the network threads when it runs with a low engine configuration on machines with a lower number of CPUs.

## Increased number of parameters

---

Set the network polling mode configuration parameter to avoid this contention. When you set network polling mode to inline, one engine performs polling requiring a separate thread. On the Linux platform, setting the network polling mode reduces CPU usage by using high-resolution sleep API.

### *network polling mode*

Summary information	
Default value	threaded
Range of values	threaded or inline
Status	Static
Display level	Basic
Required role	System administrator
Configuration group	Network Communication

When network polling mode is set to threaded, Adaptive Server spawns a separate thread for each network task configured that performs polling. When set to inline, one of the engines performs the polling.

### **Changes to *number of network tasks***

number of network tasks functions only when network polling mode is set to threaded.

## **Increased number of parameters**

Adaptive Server 15.7 ESD #1 increases from 2048 to 32767 the maximum number of parameters you can include in dynamic SQL statements and parameterized language statements.

## Multiple listeners on Windows

Computers running Microsoft Windows often have more than one network connection available. Adaptive Server versions 15.7 ESD #1 and later start listener tasks on every discrete network connection it can identify on the local computer, ensuring that your clients can connect, regardless of the physical network connection they are using.

Use `sp_listener` to alter the default listener configuration, start additional listeners, or stop existing ones, specifying the IP address of the network connection to change.

---

**Note** Make sure you use the IP address with `sp_listener` and not the host name: using the host name results in `sp_listener` affecting only the first listed network interface, which may not be the one you want.

---

Use the Windows `ipconfig` command to identify your available network connections. See your operating system documentation.

## 64-bit Windows supports high availability

Adaptive Server version 15.7 running on the 64-bit Windows operating system supports high availability. See *Using Failover in a High Availability System*.

