SQL Anywhere Monitor Non-GUI User Guide

SAP SQL Anywhere Monitor was initially implemented using Adobe Flash, which effective December 31, 2020 will be end of life. Going forward, the SAP SQL Anywhere Monitor will no longer provide a GUI interface.

This document describes how to interact with the SAP SQL Anywhere Monitor database if you need to continue using that tool for monitoring. SQL Anywhere Cockpit or SAP Solution Manager are recommended to replace monitoring requirements for SAP SQL Anywhere database servers. SAP Solution Manager also provides tools that you can use to monitor MobiLink as part of your overall SAP landscape. The MobiLink Profiler can serve to monitor a MobiLink server, but not in the same detail as the SQL Anywhere Monitor. There is no alternative tool to monitor web services.

Configuring the Monitor Database

The following section describes the procedures and tables for managing and configuring the Monitoring database to monitor SAP SQL Anywhere servers, MobiLink servers and farms, and web services.

Starting SAP SQL Anywhere Monitor

The SA Monitor database should be set up to run automatically so that it is capturing statistics and reporting alerts. Use the following configuration:

-n SAMonitor_*machine_name*
-sb 0
-ch 25p
-xd
-xs http(port=4950;MaxRequestSize=4m)
<path>\samonitor.db

The database credentials are UID: MDBA and password: sqlany.

Managing SAP SQL Anywhere Monitoring

Administration | Resources| Add | SQL Anywhere Server

add_SA_Monitor procedure

Adds a new SAP SQL Anywhere Server resource.

Syntax

add_SA_Monitor([*parameter*],[*parameter*] …)

Parameters

Parameter	Type	Description
new_monitor_name	LONG NVARCHAR	Name that identifies the SQL Anywhere Server resource

new_host_name	LONG NVARCHAR	Host name or IP address where the SAP SQL Anywhere server is running
new_server_name	LONG NVARCHAR	Name of the SAP SQL Anywhere database server
new_database_name	LONG NVARCHAR	Name of the SAP SQL Anywhere server database
new_user_name	LONG NVARCHAR	Database user ID with the SYS_SAMONITOR_ADMIN_ROLE or dba authority for version 12
new_passwd	LONG NVARCHAR	Database password
new_port	UNSIGNED INTEGER	Port that the database server is listening
new_connection_string	LONG NVARCHAR	Additional connection parameters. See Alphabetical list of connection parameters.
new_collection_rate	INTEGER	Collection interval in milliseconds
new_long_query_threshold	UNSIGNED INTEGER	Query runtime time threshold in seconds
new_conn_count_percent	UNSIGNED INTEGER	Percent of licenses in use threshold
new_blocked_conn_threshold	UNSIGNED INTEGER	Connection blocked time threshold in seconds
new_cpu_percent_threshold	UNSIGNED INTEGER	CPU usage percentage threshold
new_cpu_time_threshold	UNSIGNED INTEGER	CPU usage timeframe threshold in seconds
new_mem_percent_threshold	UNSIGNED INTEGER	Memory usage as percentage of maximum cache size threshold
new_unsched_req_threshold	UNSIGNED INTEGER	Number of unscheduled requests threshold
new_dbspace_threshold	UNSIGNED INTEGER	Free disk space per dbspace threshold in MB
new_last_backup_threshold	UNSIGNED INTEGER	Last server backup threshold in days
new_scaleout_down_threshold	UNSIGNED INTEGER	Number disconnected scale-out nodes threshold
new_alert_send_rate	UNSIGNED INTEGER	Suppress alerts for same condition within timeframe in ms (permitted values 5 minutes, 30 minutes, 1 hour, 2 hours, 6 hours, 24 hours)

Result Set

Column Name	Type	Description
MonitorID	INTEGER	Uniquely identifies the resource
ReturnCode	INTEGER	Error Status
ReturnString	LONG NVARCHAR	Error String

Default values used in the SAMonitor user interface

SA Monitor Desc	Procedure Parameter	Default
Collection Interval	new_collection_rate	30000
CPU Usage	new_cpu_percent_threshold	95
CPU Usage Time	new_cpu_time_threshold	30
Memory Usage	new_mem_percent_threshold	90
Free Disk Space	new_dbspace_threshold	100
Blocked Time	new_blocked_conn_threshold	10
Connection Count	new_long_query_threshold	85
Long Query	new_long_query_threshold	10
Unscheduled Requests	new_unsched_req_threshold	5
Last Backup	new_last_backup_threshold	14
Scaleout Down	new_alert_send_rate	25
Suppress Alerts	new_alert_send_rate	1800000

Example

```
CALL add_SA_Monitor(
        Sample ML ConsDB Server',     -- Name
        'localhost',                  -- Host
        MLConsDB', 'ConsDB,           -- ServerName, DatabaseName
        'dba', 'sql',                 -- UserName, Password
        2638,                         -- Port
        '',                           -- ConnectionString
        30000,                        -- CollectionRate
        10,                           -- LongQuery
        85,                           -- ConnectionCount
        10,                           -- BlockedTime
        95, 30,                       -- CPUUsage, CPUUsageTime
        90,                           -- MemoryUsage
        5,                            -- UnschRequests
        100,                          -- FreeDiskSpace
        14,                           -- LastBackup
        25,                           -- ScaleoutDown
        1800000 );                    -- SuppressAlerts
```

Administration | Resources| Update | <SQL Anywhere Server resource>

update_SA_MONITOR procedure

Modifies an existing SQL Anywhere Server resource.

Syntax

update_SA_MONITOR([*parameter*],[*parameter*] …)

Parameters

Parameter	Type	Description
_MonitorID	Integer	MonitorID is the value returned from add_SA_Monitor or it can be queried from the mdba.monitors table. This value must to match the monitor to be updated.
_monitor_name	LONG NVARCHAR	Name that identifies the SQL Anywhere Server resource
_host_name	LONG NVARCHAR	Host name or IP address where the SAP SQL Anywhere server is running
_server_name	LONG NVARCHAR	Name of the SAP SQL Anywhere database server
_database_name	LONG NVARCHAR	Name of the SAP SQL Anywhere server database
_port	UNSIGNED INTEGER	Port that the database server is listening
_connection_string	LONG NVARCHAR	Additional connection parameters. See Alphabetical list of connection parameters.
_collection_rate	INTEGER	Collection interval in milliseconds
_long_query_threshold	UNSIGNED INTEGER	Query runtime time threshold in seconds
_conn_count_percent	UNSIGNED INTEGER	Percent of licenses in use threshold
_blocked_conn_threshold	UNSIGNED INTEGER	Connection blocked time threshold in seconds
_cpu_percent_threshold	UNSIGNED INTEGER	CPU usage percentage threshold
_cpu_time_threshold	UNSIGNED INTEGER	CPU usage timeframe threshold in seconds
_mem_percent_threshold	UNSIGNED INTEGER	Memory usage as percentage of maximum cache size threshold
_unsched_req_threshold	UNSIGNED INTEGER	Number of unscheduled requests threshold
_dbspace_threshold	UNSIGNED INTEGER	Free disk space per dbspace threshold in MB
_last_backup_threshold	UNSIGNED INTEGER	Last server backup threshold in days
_scaleout_down_threshold	UNSIGNED INTEGER	Number disconnected scale-out nodes threshold
_alert_send_rate	UNSIGNED INTEGER	Suppress alerts for same condition within timeframe in ms (permitted values 5 minutes, 30 minutes, 1 hour, 2 hours, 6 hours, 24 hours)

Example

```
CALL update_SA_MONITOR (
        2,                              -- MonitorID
        Sample ML ConsDB Server',       -- Name
```

```
'localhost',                    -- Host
'MLConsDB', 'ConsDB',           -- ServerName, DatabaseName
2638,                           -- Port
'',                             -- ConnectionString
30000,                          -- CollectionRate
10,                             -- LongQuery
85,                             -- ConnectionCount
10,                             -- BlockedTime
95, 30,                         -- CPUUsage, CPUUsageTime
90,                             -- MemoryUsage
5,                              -- UnschRequests
100,                            -- FreeDiskSpace
14,                             -- LastBackup
25,                             -- ScaleoutDown
1800000 )                       -- SuppressAlerts
```

Administration | Resources| Delete | <SQL Anywhere Server resource>

delete_SA_monitor function

Deletes an existing SQL Anywhere Server resource.

Syntax

delete_SA_monitor([*parameter*])

Parameters

- MonitorID

  Monitor ID of the resource as defined in the monitors table

Returns

- Response < 0 indicates an error

Example

CALL delete_SA_monitor (2);

Managing MobiLink Monitoring

Administration | Resources| Add | MobiLink Server

add_ML_Monitor procedure

Adds a new MobiLink Server resource.

Syntax

add_ML_Monitor([*parameter*],[*parameter*] …)

Parameters

Parameter	Type	Description
new_monitor_name	LONG NVARCHAR	Name that identifies the MobiLink server resource
new_host_name	LONG NVARCHAR	Host name or IP address where the SAP SQL Anywhere server is running
new_port	INTEGER	Port that the MobiLink server is listening
new_enc_type	CHAR(1)	N (no encryption), R (RSA), or F(RSA FIPS 140-2 certified)
new_conn_parms	LONG NVARCHAR	Additional connection parameters to the MobiLink server
new_conn_type	CHAR(5)	HTTP, HTTPS, TCPIP, or TLS
new_user_name	LONG NVARCHAR	MobiLink user ID used to connect and collect metrics
new_passwd	LONG NVARCHAR	MobiLink password
new_high_collection_rate	INTEGER	Collection interval in milliseconds
_cpu_percent_threshold	REAL	CPU usage percentage threshold
_cpu_percent_peak_width	BIGINT	CPU usage time in seconds threshold
_disk_space_threshold	BIGINT	Free disk space for MobiLink cache threshold in bytes
_pages_used_threshold	REAL	Percentage of cache pages used threshold
_pages_locked_threshold	REAL	Percentage of locked cache pages used
_dbwait_threshold	BIGINT	Wait time for database worker thread threshold in seconds
_max_sync_time_threshold	BIGINT	Longest active synchronization time threshold in seconds
_failed_sync_threshold	BIGINT	Number of failed synchronizations threshold
_failed_sync_window_size	BIGINT	Failed synchronizations timeframe threshold in milliseconds
_error_threshold	BIGINT	Number of synchronization errors threshold
_error_window_size	BIGINT	Synchronization error timeframe threshold in milliseconds
_memory_swap_threshold	REAL	Number of pages being swapped threshold
_memory_swap_peak_width	BIGINT	Page swap rate timeframe threshold in seconds
new_alert_send_rate	BIGINT	Suppress alerts for same condition within timeframe in ms (5 minutes, 30 minutes, 1 hour, 2 hours, 6 hours, 24 hours)

Result Set

Column Name	Type	Description
MonitorID	INTEGER	Uniquely identifies the resource
ReturnCode	INTEGER	Error Status
ReturnString	LONG NVARCHAR	Error String

Defaults used in the SAMonitor user interface

SA Monitor Desc	Procedure Parameter	Default
Collection Interval	new_high_collection_rate	30000
CPU Usage	_cpu_percent_threshold	100
CPU Usage Timeframe	_cpu_percent_peak_width	300000
Free Disk Space for ML Cache	_disk_space_threshold	104857600
Cache Pages Used	_pages_used_threshold	100
Percentage Locked cache pages	_pages_locked_threshold	80
Wait for DB Worker Thread Time	_dbwait_threshold	300
Longest Active Sync Time	_max_sync_time_threshold	600
Failed Sync Count	_failed_sync_threshold	20
Failed Sync Time	_failed_sync_window_size	3600000
Number of Errors	_error_threshold	50
Number of Errors Time	_error_window_size	3600000
Pages Swapped per Second	_memory_swap_threshold	256
Pages Swapped Timeframe	_memory_swap_peak_width	120000
Suppress Alerts	new_alert_send_rate	1800000

Example

```
CALL add_ML_Monitor(
        ' Sample MLSrv 1',          -- Name
        'localhost',                -- Host_name
        41000,                      -- Port   41000,
        'N',                        -- EncType
        '',                         -- ConnectionParms
        'HTTP',                     -- ConnType
        'dba', 'sql',               -- UserName, Password
        30000,                      -- CollectionRate
        100, 300000,                -- CPUUsage, CPUUsageTime
        104857600,                  -- FreeDiskSpace
        100, 80,                    -- CachePagesUsed, LockCachePages
        300,                        -- DBWait
        600,                        -- MaxSyncTime
        20, 3600000,                -- FailedSyncs, FailedSyncTime
        50, 3600000,                -- ErrorCount, ErrorTime
```

```
    256, 120000,                -- CachePageSwap, CacheSwapTime
    1800000 );                  -- SuppressAlert
```

Administration | Resources| Update | <MobiLink Server resource>

update_ML_Monitor procedure

Modifies an existing MobiLink Server resource.

Syntax

update_ML_Monitor([*parameter*],[*parameter*] …)

Parameters

Parameter	Type	Description
_MonitorID	INTEGER	MonitorID is the value returned from add_SA_Monitor or it can be queried from the mdba.monitors table. This value must to match the monitor to be updated.
_monitor_name	LONG NVARCHAR	Name that identifies the MobiLink server resource
_host_name	LONG NVARCHAR	Host name or IP address where the SAP SQL Anywhere server is running
_user_name	LONG NVARCHAR	MobiLink user ID used to connect and collect metrics
_port	INTEGER	Port that the MobiLink server is listening
_enc_type	CHAR(1)	N (no encryption), R (RSA), or F (RSA FIPS 140-2 certified)
_conn_parms	LONG NVARCHAR	Additional connection parameters to the MobiLink server
_passwd	LONG NVARCHAR	MobiLink password
_passwd_changed	BIT	Set to 1 if the password for MobiLink UID is changed
_conn_type	CHAR(5)	HTTP, HTTPS, TCPIP, or TLS
_high_collection_rate	INTEGER	Collection interval in milliseconds
_cpu_percent_threshold	REAL	CPU usage percentage threshold
_cpu_percent_peak_width	BIGINT	CPU usage time in seconds threshold
_disk_space_threshold	BIGINT	Free disk space for MobiLink cache threshold in bytes
_pages_used_threshold	REAL	Percentage of cache pages used threshold
_pages_locked_threshold	REAL	Percentage of locked cache pages used
_dbwait_threshold	BIGINT	Wait time for database worker thread threshold in seconds
_max_sync_time_threshold	BIGINT	Longest active synchronization time threshold in seconds
_failed_sync_threshold	BIGINT	Number of failed synchronizations threshold

_failed_sync_window_size	BIGINT	Failed synchronizations timeframe threshold in milliseconds
_error_threshold	BIGINT	Number of synchronization errors threshold
_error_window_size	BIGINT	Synchronization error timeframe threshold in milliseconds
_memory_swap_threshold	REAL	Number of pages being swapped threshold
_memory_swap_peak_width	BIGINT	Page swap rate timeframe threshold in seconds
new_alert_send_rate	BIGINT	Suppress alerts for same condition within timeframe in ms (5 minutes, 30 minutes, 1 hour, 2 hours, 6 hours, 24 hours)

Example

```
CALL update_ML_Monitor(
        3,                          -- MonitorID
        ' Sample MLSrv 1',          -- Name
        'localhost',                -- Host_name
        41000,                      -- Port   41000,
        'N',                        -- EncType
        '',                         -- ConnectionParms
        'HTTP',                     -- ConnType
        'dba', 'sql',               -- UserName, Password
        30000,                      -- CollectionRate
        100, 300000,                -- CPUUsage, CPUUsageTime
        104857600,                  -- FreeDiskSpace
        100, 80,                    -- CachePagesUsed, LockCachePages
        300,                        -- DBWait
        600,                        -- MaxSyncTime
        20, 3600000,                -- FailedSyncs, FailedSyncTime
        50, 3600000,                -- ErrorCount, ErrorTime
        256, 120000,                -- CachePageSwap, CacheSwapTime
        1800000 );                  -- SuppressAlert
```

Administration | Resources| Delete | <MobiLink Server resource>

delete_ML_monitor function

Deletes an existing MobiLink Server resource.

Syntax

```
delete_ML_monitor ( [parameter] )
```

Parameters

- MonitorID Monitor ID of the resource as defined in table monitors

Returns

- Response < 0 indicates an error

Example

```
call delete_ML_Monitor( 3 )
```

Administration | Resources| Add | MobiLink Server Farm

add_ml_farm function

Adds a new MobiLink Server Farm resource.

Syntax

```
add_ml_farm (  Name  )
```

Parameters

- Name Name that identifies the MobiLink server resource

Example

```
BEGIN
DECLARE farm_id integer;
DECLARE monitor_id integer;
SET farm_id = add_ml_farm( 'Sample ML Farm' );
SELECT MonitorID INTO monitor_id FROM Monitors
   WHERE monitor_type = 'ml' and monitor_name = 'Sample MLSrv 1';
CALL add_to_ml_farm( farm_id, monitor_id );
SELECT MonitorID INTO monitor_id FROM Monitors
   WHERE monitor_type = 'ml' AND monitor_name = 'Sample MLSrv 2';
CALL add_to_ml_farm( farm_id, monitor_id );
END;
```

add_to_ml_farm function

Adds a Mobilink server to a MobiLink Server Farm

Syntax

```
add_to_ml_farm( FarmId, MobiLinkServerID )
```

Parameters

- FarmId — Returned from the add_ml_farm procedure
- SuppressAlerts — Suppress alerts for same condition within timeframe in ms (5 minutes, 30 minutes, 1 hour, 2 hours, 6 hours, 24 hours)

Example

See add_ml_farm example

Administration | Resources| Update | <MobiLink Farm resource>

update_ml_farm procedure

Modifies an existing MobiLink farm resource.

Syntax

update_ml_farm([*parameter*],[*parameter*] …)

Parameters

- MonitorID — Monitor ID of the resource as defined in table *monitors*
- FarmName — Farm name

Example

CALL update_ml_farm(6, 'Sample ML Farm');

Administration | Resources| Delete | <MobiLink Farm resource>

delete_ML_farm function

Deletes an existing MobiLink Farm resource.

Syntax

delete_ML_farm ([*parameter*])

Parameters

- MonitorID — Monitor ID of the resource as defined in the monitors table

Returns

- Response — < 0 indicates an error

Example

CALL delete_ML_farm(6);

Managing Web Server, Proxy, or Hosts Monitoring

Administration | Resources| Add | Web Service – web server, proxy, or host

add_WS_Monitor procedure

Adds a new web service resource.

Syntax

add_WS_Monitor([*parameter*],[*parameter*] …)

Parameters

Parameter	Type	Description
new_monitor_name	LONG NVARCHAR	Name that identifies the web service resource
new_service_name	LONG NVARCHAR	Description of the web service resource
new_service_type	LONG NVARCHAR	web (Web Service), proxy (Proxy), or host (Host)
new_url	LONG NVARCHAR	URL of the resource
new_proxy_host	LONG NVARCHAR	Proxy host if required
new_proxy_port	LONG NVARCHAR	Proxy port if required
new_collection_rate	INTEGER	Collection interval in milliseconds
new_timeout_interval	UNSIGNED INTEGER	Wait for a response time in seconds
new_retry_count	UNSIGNED INTEGER	Number of times to retry
new_retry_interval	UNSIGNED INTEGER	Interval between retries in seconds
new_reminder_interval	UNSIGNED INTEGER	Reminder interval in seconds
new_alert_send_rate	UNSIGNED INTEGER	Suppress alerts for same condition within timeframe in ms (5 minutes, 30 minutes, 1 hour, 2 hours, 6 hours, 24 hours)

Result Set

Column Name	Type	Description
MonitorID	INTEGER	Uniquely identifies the resource
ReturnCode	INTEGER	Error Status
ReturnString	LONG NVARCHAR	Error String

Defaults used in the SAMonitor user interface

SA Monitor Desc	Procedure Parameter	Default
Collection Interval	new_ collection_rate	30000
Response Timeout	new_timeout_interval	30
Retry Count	new_retry_count	2
Interval Between Retries	new_retry_interval	60
Suppress Alerts	new_alert_send_rate	1800000

Example

```
CALL add_WS_Monitor(
        'www.google.com',           -- new_monitor_name
        'Check google.com',         -- new_service_name
        'web',                      -- new_service_type
        'https://www.google.com',    -- new_url
        '',                          -- new_proxy_host
        '',                          -- new_proxy_port
        30000,                       -- new_collection_rate
        30,                          -- new_timeout_interval
        2,                           -- new_retry_count
        60,                          -- new_retry_interval
        60,                          -- new_reminder_interval
        1800000 );                   -- new_alert_send_rate
```

Administration | Resources| Update | <Web Service resource>

update_WS_Monitor procedure

Modifies an existing web service resource.

Syntax

update_WS_Monitor ([*parameter*],[*parameter*] …)

Parameters

SA Monitor Desc	Procedure Parameter	Default
_MonitorID	INTEGER	MonitorID is the returned from the add_ML_Monitor or can be queried from the table mdba.monitors table. This needs to match the monitor to be updated.
_monitor_name	LONG NVARCHAR	Name that identifies the web service resource
_service_name	LONG NVARCHAR	Description of the web service resource
_service_type	LONG NVARCHAR	web (Web Service), proxy (Proxy), or host (Host)

_url	LONG NVARCHAR	URL of the resource
_proxy_host	LONG NVARCHAR	Proxy host if required
_proxy_port	LONG NVARCHAR	Proxy port if required
_collection_rate	INTEGER	Collection interval in milliseconds
new_timeout_interval	UNSIGNED INTEGER	Wait for a response time in seconds
new_retry_count	UNSIGNED INTEGER	Number of times to retry
new_retry_interval	UNSIGNED INTEGER	Interval between retries in seconds
new_reminder_interval	UNSIGNED INTEGER	Reminder interval in seconds
new_alert_send_rate	UNSIGNED INTEGER	Suppress alerts for same condition within timeframe in ms (5 minutes, 30 minutes, 1 hour, 2 hours, 6 hours, 24 hours)

Example

```
CALL update_WS_Monitor (
        7,                          --_MonitorID
        'www.google.com',           --_monitor_name
        'Check google.com',         --_service_name
        'web',                      --_service_type
        'https://www.google.com',   --_url
        '',                         --_proxy_host
        '',                         --_proxy_port
        30000,                      --_collection_rate
        30,                         --_timeout_interval
        2,                          --_retry_count
        60,                         --_retry_interval
        60,                         --_reminder_interval
        1800000  );                 --_alert_send_rate
```

Administration | Resources| Delete | < Web Service resource>

delete_WS_monitor function

Deletes an existing web service resource.

Syntax

delete_WS_monitor ([*parameter*])

Parameters

- MonitorID Monitor ID of the resource as defined in the monitors table

Returns

- Response < 0 indicates an error

Example

```
CALL delete_ML_monitor( 7 );
```

Managing Custom Metrics

Administration | Resources | Configuration | Custom Metrics

custom_metrics table

Defines custom metrics for the monitor resource.

Table Schema

Column Name	Type	Description
monitor_id	INTEGER	MonitorID for SA resource
name	VARCHAR(128)	Name of custom metric
data_type	VARCHAR(128)	Datatype string 'integer', 'float'
display_units	VARCHAR(128)	Description of the custom metric
minimum	DOUBLE	Minimum value of metric
maximum	DOUBLE	Maximum value of metric
alert_threshold	DOUBLE	Value to report alert
alert_level	INTEGER	0 (Low), 1 (Med), 2 (High)
alert_on_exceed_threshold	BIT	Enable the alert when exceeded
value_function_owner	VARCHAR(128)	Function owner (sa_monitor_user)
value_function_name	VARCHAR(128)	Function name
data_table_name	VARCHAR(128)	Table to store the custom metrics in the form {moniter_id}_{name}

Usage

Custom metrics can be any integer or float user specified value maintained in the database and queried by the monitor by using a function that returns its current value.

Insert a row into the custom metrics table for the monitor (based on monitor_id) that defines the metric name, display unit, min and max values, data type and the owner and name of the stored function that supplies its current value.

Example

A full example of custom metrics is provided Example Setup section of this document.

Managing Monitoring Configuration

This section discusses the configuration as blackout periods, alerts, and maintenance of the SAP SQL Anywhere Monitor database.

Administration | Resources | <Resource> | Blackouts

blackout_periods table

Defines blackout periods for the monitored resource.

Table Schema

Column Name	Type	Description
monitor_id	INTEGER	MonitorID for SA resource
blackout_start	TIME	Blackout start time
blackout_end	TIME	Blackout start time
everyDay	BIT	Set to 1 to enable blackout Every Day
daysOfTheWeek	BIT	Set to 1 to enable blackout for specific days of the week
sunday	BIT	Set to 1 to enable blackout on Sunday
monday	BIT	Set to 1 to enable blackout on Monday
tuesday	BIT	Set to 1 to enable blackout on Tuesday
wednesday	BIT	Set to 1 to enable blackout on Wednesday
thursday	BIT	Set to 1 to enable blackout on Thursday
friday	BIT	Set to 1 to enable blackout on Saturday
saturday	BIT	Set to 1 to enable blackout on Sunday
daysOfTheMonth	BIT	Set to 1 to enable blackout for specific days of the month
day1	BIT	Set to 1 to enable blackout on day 1 of Month
day2	BIT	Set to 1 to enable blackout on day 2 of Month
day3	BIT	Set to 1 to enable blackout on day 3 of Month
day4	BIT	Set to 1 to enable blackout on day 4 of Month
day5	BIT	Set to 1 to enable blackout on day 5 of Month
day6	BIT	Set to 1 to enable blackout on day 6 of Month
day7	BIT	Set to 1 to enable blackout on day 7 of Month
day8	BIT	Set to 1 to enable blackout on day 8 of Month
day9	BIT	Set to 1 to enable blackout on day 9 of Month
day10	BIT	Set to 1 to enable blackout on day 10 of Month
day11	BIT	Set to 1 to enable blackout on day 11 of Month
day12	BIT	Set to 1 to enable blackout on day 12 of Month
day13	BIT	Set to 1 to enable blackout on day 13 of Month
day14	BIT	Set to 1 to enable blackout on day 14 of Month
day15	BIT	Set to 1 to enable blackout on day 15 of Month
day16	BIT	Set to 1 to enable blackout on day 16 of Month
day16	BIT	Set to 1 to enable blackout on day 17 of Month
day18	BIT	Set to 1 to enable blackout on day 18 of Month

day19	BIT	Set to 1 to enable blackout on day 19 of Month
day20	BIT	Set to 1 to enable blackout on day 20 of Month
day21	BIT	Set to 1 to enable blackout on day 21 of Month
day22	BIT	Set to 1 to enable blackout on day 22 of Month
day23	BIT	Set to 1 to enable blackout on day 23 of Month
day24	BIT	Set to 1 to enable blackout on day 24 of Month
day25	BIT	Set to 1 to enable blackout on day 25 of Month
day26	BIT	Set to 1 to enable blackout on day 26 of Month
Day27	BIT	Set to 1 to enable blackout on day 27 of Month
day28	BIT	Set to 1 to enable blackout on day 28 of Month
day29	BIT	Set to 1 to enable blackout on day 29 of Month
day30	BIT	Set to 1 to enable blackout on day 30 of Month
day31	BIT	Set to 1 to enable backout on day 31 of Month
lastDayOfTheMonth	BIT	Set to 1 to enable blackout on last day of the month

Usage

Insert a row for the Monitor (based on Monitor ID) that specifies the start and end time of the blackout period and defines which days that the backout period applies. You can set this to be every day, specific days of the week, or specific days of the month. Setting the bit columns to 1 enables the blackout period. For example, set the "everyday" column to one to set the blackout period to every day.

Example

- Set a black period for 9pm-11pm every day

```
INSERT INTO "mdba"."blackout_periods"
        ( "MonitorID", "blackout_start", "blackout_end", "everyday" )
VALUES
        ( 3, '21:00', '23:00', 1 );
```

Administration | Resources | Configuration | Alert Notification

email_config table

Defines the email configuration for the reporting alerts. The table contains a single row as a placeholder. To configure email alerts, updated the table with the configuration for either a MAPI or SMTP mail server.

Table Schema

Column Name	Type	Description
configID	INTEGER	There is an existing row with configID 0. This row should be used for email configuration.

emailAlerts	BIT	Set 1 to enable alerts
isMAPI	BIT	Set 1 if configuration is for MAPI
mapiUsername	LONG NVARCHAR	MAPI Username
mapiPassword	LONG NVARCHAR	MAPI Password
SmtpServer	LONG NVARCHAR	SMTP Server
smtpPort	INTEGER	SMTP Port
smtpSenderName	LONG NVARCHAR	SMTP Sender Name
smtpSenderAddress	LONG NVARCHAR	SMTP Sender Address
smtpAuthenticationRequired	BIT	Set 1 if SMPT authentication is required
SmtpUsername	LONG NVARCHAR	SMTP Username
smtpPassword	LONG NVARCHAR	SMTP Password
secureSMTPRequired	BIT	Set 1 if SMTP requires TLS connection
smtpTrustedCertificates	LONG NVARCHAR	SMTP TLS trusted certificates
smtpCertificateCompany	LONG NVARCHAR	SMTP TLS Certificate Company--set NULL if not used
smtpCertificateUnit	LONG NVARCHAR	SMTP TLS Certificate Unit--set NULL if not used
smtpCertificateName	LONG NVARCHAR	SMTP TLS Certificate Name--set NULL if not used
onlyEmailHighPriority	BIT	Set 1 to send only high priority alerts
limitEmailsSent	BIT	Set 1 to limit emails sent per day
emailLimit	INTEGER	Max number of emails per day

Example

```
UPDATE "mdba"."email_config"
SET
        "configID" = 0,
        "emailAlerts" = 1,
        "smtpServer" = 'corporatemail.sample.com',
        "smtpPort" = 25,
        "smtpSenderName" = 'Jane Doe',
        "smtpSenderAddress" = 'jdoe@sample.com'
WHERE "configID" = 0 ;
```


update_maintenance_config procedure

Updates the maintenance configuration of the Monitor. This procedure performs a backup of the Monitor database and deletes statistics that are older than a specified number of days.

Syntax

update_maintenace_config ([*parameter*])

Parameters

Parameter	Type	Description
new_maint_time	TIME	The time when maintenance will be performed daily
new_backup_location	LONG NVARCHAR	The directory, which should be on the same computer that the SAP SQL Anywhere Monitor is running, where to the SQL Anywhere Monitor database is being backed up
new_do_avg	BIT	Set to 1 to enable calculation of averages
new_num_days_old_for_avg	INTEGER	Set to 14
new_do_delete	BIT	Set to 1 to enable cleanup of statistics
new_num_days_old_for_delete	UNSIGNED INTEGER	Set to 5
new_do_file_size_delete	BIT	Set to 1 to enable
new_file_size_for_delete	UNSIGNED INTEGER	Set to 1024
new_pct_to_keep_free	UNSIGNED INTEGER	Set to 20
new_num_days_for_file_delete	UNSIGNED INTEGER	Set to 1

Example

```
CALL update_maintenance_config (
        new_maint_time = '02:30',
        new_backup_location = 'c:/backup',
        new_do_avg = 0,
        new_num_days_old_for_avg = 14,
        new_do_delete = 0,
        new_num_days_old_for_delete = 7,
        new_do_file_size_delete = 0,
        new_file_size_for_delete = 1024,
        new_pct_to_keep_free = 20,
        new_num_days_for_file_delete = 1);
```

Reporting on Monitored Statistics

This section describes the tables that contain the monitoring statistics that are collected and how to report on those statistics.

message_log table

Stores messages logged by the SAP SQL Anywhere monitor for configured monitored resources.

Schema

- message_id INTEGER Message identifier
- message_text LONG NVARCHAR Text of the message
- message_time TIMESTAMP Timestamp of the logged message
- monitorID INTEGER Monitored resource of the message
- severity INTEGER Severity of the message (High=3, Medium=2, Low=1)

Example:

A monitor resource named "SQL Anywhere Demo 17" has been configured for the SQL Anywhere 17 demo database. To retrieve the messages for that resource, execute

```
SELECT
   message_time, message_text, severity
FROM
   message_log
WHERE
   monitorID = (
      SELECT
         MonitorID
      FROM
         monitors
      WHERE
         monitor_name = 'SQL Anywhere Demo 17' )
ORDER BY message_time;
```

SA_conn_and_seat_count table

Stored connection and seat count statistics for configured monitored resources.

Schema

• conn_count_pkey	INTEGER	Connection count statistic identifier
• MonitorID	INTEGER	Monitored resource for the connection count statistic
• collection_time	TIMESTAMP	Timestamp of the connection count statistic
• conn_count	UNSIGNED INT	Connection count
• seat_count	UNSIGNED INT	Seat count
• data_start	BIT	1 if the statistic is first after monitor started
• averaged	BIT	Not used

Example

A monitor resource named "SQL Anywhere Demo 17" has been configured for the SQL Anywhere 17 demo database. To retrieve the seat and connection statistics for that resource, execute

```
SELECT
   conn_count, seat_count
FROM
   SA_conn_and_seat_count
WHERE
   MonitorID = (
      SELECT
         MonitorID
      FROM
         monitors
      WHERE
         monitor_name = 'SQL Anywhere Demo 17' );
```

SA_conn_failure table

Stored connection failure occurrences.

Schema

- connection_failure_id INTEGER Connection failure statistic identifier
- MonitorID INTEGER Monitored resource for the connection failure statistic
- user_name CHAR(128) User id associated with the failed connection
- app_info LONG NVARCHAR AppInfo for the failed connection
- collection_time TIMESTAMP Timestamp of the failed connection

Example

A monitor resource named "SQL Anywhere Demo 17" has been configured for the SQL Anywhere 17 demo database. To retrieve failed connection information for that resource, execute

```
SELECT
   user_name, app_info, collection_time
FROM SA_conn_failure
WHERE
   MonitorID = (
      SELECT MonitorID
      FROM monitors
      WHERE
monitor_name = 'SQL Anywhere Demo 17' );
```

SA_cpu_usage table

Stored CPU usage statistics for configured monitored resources.

Schema

- cpu_usage_pkey INTEGER CPU usage statistic identifier
- MonitorID INTEGER Monitored resource for the CPU usage statistic
- collection_time TIMESTAMP Timestamp for the CPU usage statistic
- cpu_percent UNSIGNED INT CPU usage percentage
- data_start BIT 1 if the statistic is first after monitor started
- averaged BIT Not used

Example

A monitor resource named "SQL Anywhere Demo 17" has been configured for the SQL Anywhere 17 demo database. To retrieve CPU usage information for that resource, execute

```
SELECT
   collection_time, cpu_percent, data_start
```

```
FROM SA_cpu_usage
WHERE
   MonitorID = (
      SELECT MonitorID
      FROM monitors
      WHERE
monitor_name = 'SQL Anywhere Demo 17' );
```

SA_dbspace table

Stored dbspace statistics for configured monitored resources

Schema

- dbspace_pkey INTEGER DBSpace statistic identifier
- MonitorID INTEGER Monitored resource for the dbspace statistic
- collection_time TIMESTAMP Timestamp for the dbspace statistic
- dbSpaceName LONG NVARCHAR Name of the dbspace
- file_size UNSIGNED BIGINT Size of the dbspace
- free_space UNSIGNED BIGINT Size of the free sapce
- total_space UNSIGNED BIGINT Size of the total space
- data_start BIT 1 if the statistic is first after monitor started
- averaged BIT Not used

Example

A monitor resource named "SQL Anywhere Demo 17" has been configured for the SQL Anywhere 17 demo database. To retrieve the database disk space statistics for that resource, execute

```
SELECT
   collection_time, dbSpaceName,
   file_size, free_space, total_space,
   data_start
FROM SA_dbspace
WHERE
   MonitorID = (
      SELECT MonitorID
      FROM monitors
      WHERE
monitor_name = 'SQL Anywhere Demo 17' );
```

SA_disk_io table

Stored disk I/O statistics for configured monitored resources

Schema

- disk_io_pkey INTEGER Disk IO statistic primary key
- MonitorID INTEGER Monitored resource for the Disk IO statistic

- collection_time TIMESTAMP Timestamp for the Disk IO statistic
- disk_read_rate FLOAT Disk read rate
- disk_written_rate FLOAT Disk write rate
- data_start BIT 1 if the statistic is first after monitor started
- averaged BIT Not used

Example

A monitor resource named "SQL Anywhere Demo 17" has been configured for the SQL Anywhere 17 demo database. To retrieve the disk IO statistics for that resource, execute

```
SELECT
   collection_time, disk_read_rate, disk_written_rate, data_start
FROM SA_disk_io
WHERE
   MonitorID = (
      SELECT MonitorID
      FROM monitors
      WHERE
monitor_name = 'SQL Anywhere Demo 17' );
```

SA_http_info table

Stored HTTP info statistics for configured monitored resources

Schema

- http_info_pkey INTEGER HTTP statistic primary key
- MonitorID INTEGER Monitored resource for the HTTP statistic
- collection_time TIMESTAMP Timestamp for the HTTP statistic
- http_type UNSIGNED INT Type of HTTP statistic collected (see below).
- value UNSIGNED INT Value of the statistic
- data_start BIT 1 if the statistic is first after monitor started
- averaged BIT Not used

http_type values

- HttpNumConnections 1
- HttpsNumConnections 2
- HttpNumActiveReq 3
- HttpsNumActiveReg 4
- HttpNumSessions 5

Example

A monitor resource named "SQL Anywhere Demo 17" has been configured for the SQL Anywhere 17 demo database. To retrieve the HTTP info statistics for that resource, execute

```
SELECT
   collection_time, http_type, value, data_start
FROM SA_http_info
```

```
WHERE
   MonitorID = (
      SELECT MonitorID
      FROM monitors
      WHERE
monitor_name = 'SQL Anywhere Demo 17' );
```

SA_long_query_info table

Stored long query statistics for configured monitored resources

Schema

- long_query_pkey INTEGER Long query statistic identifier
- MonitorID INTEGER Monitored resource for the long query statistic
- collection_time TIMESTAMP Timestamp for the long query statistic
- conn_number UNSIGNED INT SQL Anywhere connection identifier
- conn_name LONG NVARCHAR SQL Anywhere connection name
- duration UNSIGNED INT Duration of the long query
- statement_text LONG NVARCHAR SQL statement of the long query

Example

A monitor resource named "SQL Anywhere Demo 17" has been configured for the SQL Anywhere 17 demo database. To retrieve the long query statistics for that resource, execute

```
SELECT
  collection_time, conn_number, conn_name,
  duration,
  statement_text
FROM SA_long_query_info
WHERE
   MonitorID = (
      SELECT MonitorID
      FROM monitors
      WHERE
monitor_name = 'SQL Anywhere Demo 17' );
```

SA_mem_usage table

Stored memory usage statistics for configured monitored resources

Schema

- mem_usage_pkey INTEGER Long query statistic identifier
- MonitorID INTEGER Monitored resource for the long query statistic
- collection_time TIMESTAMP Timestamp for the long query statistic
- mem_type UNSIGNED INT Memory type
- value UNSIGNED BIGINT Value of the statistic
- data_start BIT 1 if the statistic is first after monitor started

- averaged BIT Not used

mem_type values

- CurrentCacheSize 1
- MainHeapPages 2
- PeakCacheSize 3
- CachePinned 4
- CacheFileDirty 5
- CacheReplacements 6

Example

A monitor resource named "SQL Anywhere Demo 17" has been configured for the SQL Anywhere 17 demo database. To retrieve the memory usage statistics for that resource, execute

```
SELECT
   collection_time, mem_type, value, data_start
FROM SA_mem_usage
WHERE
   MonitorID = (
      SELECT MonitorID
      FROM monitors
      WHERE
monitor_name = 'SQL Anywhere Demo 17' );
```

SA_mirror_info table

Stored mirror information statistics for configured monitored resources

Schema

- mirror_info_pkey INTEGER Mirroring statistic identifier
- MonitorID INTEGER Monitored resource for the mirroring statistic
- collection_time TIMESTAMP Timestamp for the mirroring statistic
- mirror_mode NVARCHAR(32) MirrorMode database property value
- mirror_state NVARCHAR(32) MirrorState database property value
- partner_state NVARCHAR(32) PartnerState database property value
- arbiter_state NVARCHAR(32) ArbiterState database property value
- data_start BIT 1 if the statistic is first after monitor started

Example

A monitor resource named "SQL Anywhere Demo 17" has been configured for the SQL Anywhere 17 demo database. To retrieve the mirror information statistics for that resource, execute

```
SELECT
   collection_time,
   mirror_mode, mirror_state,
   partner_state, arbiter_state,
```

```
   data_start
FROM SA_mirror_info
WHERE
   MonitorID = (
      SELECT MonitorID
      FROM monitors
      WHERE
monitor_name = 'SQL Anywhere Demo 17' );
```

SA_monitor_info table

Stored monitor information snapshot for configured monitored resources

Schema

- MonitorID — INTEGER
  Monitored resource for the monitor information
- last_row_update — TIMESTAMP
  Update time for the monitor information
- respone_time_mls — UNSIGNED INT
  Not Used.
- server_name — LONG NVARCHAR
  SQL Anywhere server name
- database_name — LONG NVARCHAR
  SQL Anywhere database name
- Version — LONG NVARCHAR
  SQL Anywhere version details
- iq_server — BIT
  1 if the server is IQ
- network_server — BIT
  1 if the server is a network server
- personal_server — BIT
  1 if the server is a personal server
- language — LONG NVARCHAR
  Language database property
- start_time — LONG NVARCHAR
  Start time of the database server
- license_type — LONG NVARCHAR
  Server license type
- license_number_of_seats — INTEGER
  Server license count (Seats or Cores)
- license_company_name — LONG NVARCHAR
  License company name
- license_user_name — LONG NVARCHAR
  License user name
- host_name — LONG NVARCHAR
  Database server host name

- host_operating_system — LONG NVARCHAR
  Host operating system
- host_operating_system_version — LONG NVARCHAR
  Host operating system version
- host_processor_architecture — LONG NVARCHAR
  Host operating system architecture
- max_cache_size — UNSIGNED BIGINT
  Maximum cache size
- min_cache_size — UNSIGNED BIGINT
  Minimum cache size
- page_size — UNSIGNED BIGINT
  Database page size
- critical_update_avail — BIT
  Critical software update available
- other_update_avail — BIT
  Software update available
- unsubmitted_error_report_count — UNSIGNED INT
  Unsubmitted error report count
- http_server_enabled — BIT
  1 if the HTTP is enabled
- last_backup_time — TIMESTAMP
  Last backup time
- time_zone — INTEGER
  Timezone of the server
- httpAddresses — LONG NVARCHAR
  Configured HTTP addresses
- httpsAddresses — LONG NVARCHAR
  Configured HTTPS addresses

Example

A monitor resource named "SQL Anywhere Demo 17" has been configured for the SQL Anywhere 17 demo database. To retrieve the current monitor info snapshot for that resource, execute

```
SELECT
  *
FROM SA_monitor_info
WHERE
  MonitorID = (
    SELECT MonitorID
    FROM monitors
    WHERE
monitor_name = 'SQL Anywhere Demo 17' );
```

SA_queries_processed table

Stored queries processed statistics for configured monitored resources

Schema

• queries_processed_pkey	INTEGER	Queries processed statistic identifier
• MonitorID	INTEGER	Monitored resource for the queries processed statistic
• collection_time	TIMESTAMP	Timestamp for the queries processed statistic
• queries_processed_rate	FLOAT	Queries processed rate
• data_start	BIT	1 if the statistic is first after monitor started
• averaged	BIT	Not used

Example

A monitor resource named "SQL Anywhere Demo 17" has been configured for the SQL Anywhere 17 demo database. To retrieve the queries processed statistics for that resource, execute

```
SELECT
  collection_time, queries_processed_rate, data_start
FROM SA_queries_processed
WHERE
   MonitorID = (
      SELECT MonitorID
      FROM monitors
      WHERE
monitor_name = 'SQL Anywhere Demo 17' );
```

SA_unsched_requests table

Stored unscheduled requests statistics for configured monitored resources

Schema

• unsch_req_pkey	INTEGER	Unscheduled requests statistic identifier
• MonitorID	INTEGER	Monitored resource for the unscheduled Requests statistic
• collection_time	TIMESTAMP	Timestamp for the unscheduled requests statistics
• num_unsched_requests	UNSIGNED INT	Unscheduled requests statistic
• data_start	BIT	1 if the statistic is first after monitor started
• averaged	BIT	Not used

Example

A monitor resource named "SQL Anywhere Demo 17" has been configured for the SQL Anywhere 17 demo database. To retrieve the current monitor info snapshot for that resource, execute

```
SELECT
  collection_time, num_unsched_requests, data_start
```

```
FROM SA_unsched_requests
WHERE
   MonitorID = (
      SELECT MonitorID
      FROM monitors
      WHERE
monitor_name = 'SQL Anywhere Demo 17' );
```

SA_unsub_error_rpt table

Stored unsubmitted error report statistics for configured monitored resources

Schema

err_rpt_pkey	INTEGER
MonitorID	INTEGER
collection_time	TIMESTAMP
num_unsub_rpts	UNSIGNED INT
data_start	BIT

Example

A monitor resource named "SQL Anywhere Demo 17" has been configured for the SQL Anywhere 17 demo database. To retrieve the current monitor info snapshot for that resource, execute

```
SELECT
  *
FROM SA_monitor_info
WHERE
   MonitorID = (
      SELECT MonitorID
      FROM monitors
      WHERE
monitor_name = 'SQL Anywhere Demo 17' );
```

SA_monitor_info table

Stored monitor information snapshot for configured monitored resources

MobiLink BlackBoard Value Types

Example Setup

This example will set up monitoring of a consolidated database running in a MobiLink Farm consisting of two nodes connecting over TCP/IP. Further, you will monitor the health of the host system.

Set up the Consolidated Database for Monitoring

The Monitor resource name will be Sample ML ConsDB Server. The database is running on LocalHost on the default port with a Server Name of MLConsDb and a database name of ConsDB. We will accept the default monitoring Alert settings. To add this to the SQL Anywhere Monitor, execute from DBISQL:

```
CALL add_SA_Monitor(
        'Sample ML ConsDB Server',      -- Name
        'localhost',                    -- Host
        'MLConsDB', 'ConsDB',           -- ServerName, DatabaseName
        'dba', 'sql',                   -- UserName, Password
        2638,                           -- Port
        '',                             -- ConnectionString
        30000,                          -- CollectionRate
        10,                             -- LongQuery
        85,                             -- ConnectionCount
        10,                             -- BlockedTime
        95, 30,                         -- CPUUsage, CPUUsageTime
        90,                             -- MemoryUsage
        5,                              -- UnschRequests
        100,                            -- FreeDiskSpace
        14,                             -- LastBackup
        25,                             -- ScaleoutDown
        1800000 );              -- SuppressAlerts
```

This will return the new Monitor ID if successful. A negative Monitor ID is returned in the case of a failure along with a return code and string. For this example, the call returns a new MonitorID = 2.

To determine the MonitorID at a later point, query the Monitors table. For example

```
SELECT
        monitorID
FROM
        monitors
WHERE
        monitor_type = 'sa' AND
        monitor_name = 'Sample ML ConsDB Server';
```

Set up the MobiLink Servers for Monitoring

There will be two MobiLink Servers that will be participating in a MobiLink farm. The connections will be TCP/IP based. The first MobiLink server will be running on port 41000 and the second on port 42000. The default alert configuration settings will be used.

To add the first server to the Monitor, execute

```
CALL add_ML_Monitor(
        'Sample MLSrv 1',           -- Name
        'localhost',                -- Host_name
        41000,                      -- Port
        'N',                        -- EncType
        '',                         -- ConnectionParms
        'TCPIP',                    -- ConnType
        'dba', 'sql',               -- UserName, Password
        30000,                      -- CollectionRate
        100, 300000,                -- CPUUsage, CPUUsageTime
        104857600,                  -- FreeDiskSpace
        100, 80,                    -- CachePagesUsed, LockCachePages
        300,                        -- DBWait
        600,                        -- MaxSyncTime
        20, 3600000,                -- FailedSyncs, FailedSyncTime
        50, 3600000,                -- ErrorCount, ErrorTime
        256, 120000,                -- CachePageSwap, CacheSwapTime
        1800000 );                  -- SuppressAlert
```

This will return the new Monitor ID if successful. A negative Monitor ID is returned in the case of a failure along with a return code and string. For this example, the call returns a new MonitorID = 3.

To determine the MonitorID at a later point, query the Monitors table. For example

```
SELECT
        monitorID
FROM
        monitors
WHERE
        monitor_type = 'ml' AND
        monitor_name = 'Sample MLSrv 1';
```

To add the second server to the Monitor, execute

```
CALL add_ML_Monitor(
        'Sample MLSrv 2',           -- Name
        'localhost',                -- Host_name
        42000,                      -- Port
        'N',                        -- EncType
        '',                         -- ConnectionParms
        'TCPIP',                    -- ConnType
        'dba', 'sql',               -- UserName, Password
        30000,                      -- CollectionRate
        100, 300000,                -- CPUUsage, CPUUsageTime
```

```
104857600,              -- FreeDiskSpace
100, 80,                -- CachePagesUsed, LockCachePages
300,                    -- DBWait
600,                    -- MaxSyncTime
20, 3600000,            -- FailedSyncs, FailedSyncTime
50, 3600000,            -- ErrorCount, ErrorTime
256, 120000,            -- CachePageSwap, CacheSwapTime
1800000 );              -- SuppressAlert
```

For this example, the call returns a new MonitorID = 4.

Set up the MobiLink Farm for Monitoring

The *Sample MLSrv 1* and *Sample MLSrv 2* will be participating in the MobiLink Farm *Sample ML Farm*. To set this farm, execute

```
BEGIN
DECLARE farm_id INTEGER;
DECLARE monitor_id INTEGER;
SET farm_id = add_ml_farm( 'Sample ML Farm' );
SELECT MonitorID INTO monitor_id FROM Monitors
   WHERE monitor_type = 'ml' AND monitor_name = 'Sample MLSrv 1';
CALL add_to_ml_farm( farm_id, monitor_id );
SELECT MonitorID INTO monitor_id FROM Monitors
   WHERE monitor_type = 'ml' AND monitor_name = 'Sample MLSrv 2';
CALL add_to_ml_farm( farm_id, monitor_id );
END;
```

The add_ml_farm is a function and returns MonitorID. For this example, the new MonitorID = 5

To determine the MobiLink Farm MonitorID at a later point, query the Monitors table. For example

```
SELECT
        monitorID
FROM
        monitors
WHERE
        monitor_type = 'mf' AND
        monitor_name = 'Sample ML Farm';
```

Set up the Host System for Monitoring

It is sometimes helpful to distinguish between the applications being down versus the host machine for the applications. To set up to monitor the Host machine, add WS monitoring of the host. In this example:

```
CALL add_WS_Monitor(
        'localhost',                    -- new_monitor_name
        'Monitor localhost',            -- new_service_name
        'host',                         -- new_service_type
        'localhost',                    -- new_url
        '',                              -- new_proxy_host
        '',                              -- new_proxy_port
        30000,                           -- new_collection_rate
        30,                              -- new_timeout_interval
        2,                               -- new_retry_count
        60,                              -- new_retry_interval
        60,                              -- new_reminder_interval
        1800000 );                       -- new_alert_send_rate
```

This will return the new Monitor ID if successful. A negative Monitor ID is returned in the case of a failure along with a return code and string. For this example, the call returns a new MonitorID = 6.

To determine the MonitorID at a later point, query the Monitors table. For example

```
SELECT
        monitorID
FROM
        monitors
WHERE
        monitor_type = 'ws' AND
        monitor_name = 'localhost';
```

Creating and Managing Custom Metrics

The monitor has the capability to manage user defined custom metrics on a monitored SAP SQL Anywhere database. To implement this functionality, create a function in the monitored SAP SQL Anywhere database that returns either an integer or float based value. In this example, the DBA is interested in monitoring the number of local connections to the database server that are not internal connections such as events. The function that is defined is

```
CREATE OR REPLACE FUNCTION sa_monitor_user.numberLocalConnections()
        RETURNS INTEGER
BEGIN
        DECLARE numConns INTEGER;
        SELECT count(*) INTO numConns FROM sa_conn_info() WHERE NUMBER < 1000000000;
        RETURN numConns
END;
```

The user is set to sa_monitor_user to simplify access to the function by the monitor. Alternatively, you would permit sa_monitor_user to execute the function.

Assume that the SA Monitor resource is Sample ML ConsDB Server which has the MonitorID 2.

```
INSERT INTO "mdba"."custom_metrics" (
        "monitor_id",
        "name",
        "data_type",
        "display_units",
        "minimum",
        "maximum",
        "alert_threshold",
        "alert_level",
        "alert_on_exceed_threshold",
        "value_function_owner",
        "value_function_name",
        "data_table_name" )
VALUES (
        2,
        'NumberLocalConnections',
        'integer',
        'Local connection count',
        0,
        999999999,
        50,
        2,
        1,
        'sa_monitor_user',
        'numberLocalConnections',
        '2_NumberLocalConnections' );
```

To store the custom metrics, you must create a table. The table name will be {*monitor_id*}_{*name*} and a primary key {*monitor_id*}_{*name*}_pkey. In this example, the monitor_id is 2 and the name is 'NumberLocalConnections' so the resulting table will be:

```
CREATE TABLE mdba."2_NumberLocalConnections" (
        "2_NumberLocalConnections_pkey"     INTEGER NOT NULL DEFAULT AUTOINCREMENT,
        "value"                             INTEGER NULL,
        "collection_time"                   TIMESTAMP NULL,
        "data_start"                        BIT NOT NULL,
        PRIMARY KEY ( "2_NumberLocalConnections_pkey" ASC ) );
```

Configuring Users for Email Alerts

users table

The SAP SQL Anywhere Monitor GUI provided application-level logins. With the non-GUI Monitor, the only purpose of defining users is to define the email addresses for email alerts. If you require similar user permissions of that provided at the application level, you can create roles for Administrator, Operator, and Read-only user that provide similar permissions to the database.

For email notifications, add a new user to the users table and then subscribe the user to the Monitor resource by adding a row to the monitor_operators table which as the users table userID and monitors table MonitorID.

Schema

• userID	INTEGER	Identifies the user row
• UserName	LONG NVARCHAR	Application level user name
• Password	LONG NVARCHAR	Application level password
• user_type	INTEGER	1 (Administrator), 2 (Operator), 3 (Read-Only user)
• email	LONG NVARCHAR	Email address used for Alert emails
• language	CHAR(2)	en (English), de (German), fr (French), ja (Japanese), zh (Simplified Chinese)
• update_interval	INTEGER	Update frequency in ms (default 60000)
• ui_definition	LONG NVARCHAR	Not Used
• num_emails_today	INTEGER	Count of emails sent to user today

To create a user named mon_user that is an Operator, execute the following:

```
INSERT INTO "mdba"."users"(
        "userName",
        "password",
        "user_type",
        "email",
        "language",
        "update_interval",
        "ui_definition",
        "num_emails_today" )
VALUES (
        'mon_user',
        '',
        2,
        '<user_email_address>',
        'en',
        60000,
        '',
        0 );
```

The password that would be provided to the user interface and is not provided in this context. Do not include the real password as it will be stored in plain text.

For each Monitor in the monitors table that the user wants to receive alerts for, add a new row to the monitor_operators table which is a pairing of MonitorID and userId primary key values. In the example setup, we have created monitors for the SAP SQL Anywhere Monitor, Sample ML ConsDB Server, Sample MLSrv 1 and 2, Sample ML farm, and the host localhost with MonitorIDs 1 through 7 The mon_user userID is 2. To get alerts for the monitored resources, insert the following rows:

```
INSERT INTO "mdba"."monitor_operators" ( "MonitorID", "userID" ) VALUES( 1, 2);
INSERT INTO "mdba"."monitor_operators" ( "MonitorID", "userID" ) VALUES( 2, 2);
INSERT INTO "mdba"."monitor_operators" ( "MonitorID", "userID" ) VALUES( 3, 2);
INSERT INTO "mdba"."monitor_operators" ( "MonitorID", "userID" ) VALUES( 4, 2);
INSERT INTO "mdba"."monitor_operators" ( "MonitorID", "userID" ) VALUES( 6, 2);
INSERT INTO "mdba"."monitor_operators" ( "MonitorID", "userID" ) VALUES( 7, 2);
```

MobiLink BlackBoard Value Types

Type Value	Statistic Description
-34	Started
-2	Completed Synchronization Rate
-3	Failed Synchronization Rate
-35	Finished Synchronization Rate
-4	Synchronization Error Rate
-5	Synchronization Warning Rate
37	Max Synchronization Time
-31	Requests
41	Synchronizations
42	Pings
43	File Transfers
44	Monitors
45	Listeners
-24	Unknown Connected Clients
-28	Authenticating Synchronizations
-29	Uploading Synchronizations
-30	Downloading Synchronizations
-22	Percentage of Connections in Use
62	Upload Connections in Use
-27	Synchronizations Blocked
36	Max Wait for Connection
-9	Commit Rate

-10	Rollback Rate
46	Waiting Connections
-6	Rows Downloaded Rate
-7	Rows Uploaded Rate
-8	Percentage of Clients Connected
20	CPU Total Time
-1	CPU Usage
4	Server Cache Size
39	Memory Used
-11	Percent of Pages Used
-12	Percent of Pages Locked
-14	Pages Swapped Out Rate
-15	Pages Swapped In Rate
64	VM Memory Usage
0	Current TCP Connections
-16	TCP Bytes Read Rate
-17	TCP Bytes Written Rate
-20	Connections Rejected Rate
35	Free Disk Space for MobiLink Cache
67	Cached Remote Task Requests
-42	RTNotifier Lightweight Poll Rate
-41	RTNotifier Lightweight Poll Miss Rate
-40	RTNotifier Lightweight Poll Hit Rate
48	Requests in Synchronization Request Phase
49	Requests in Receive Upload phase
46	Requests in Get DB Worker Phase
50	Requests in Connect Phase
-26	Requests in Authenticate User Phase
52	Requests in Begin Synchronization Phase
53	Requests in Apply Upload Phase
54	Requests in Prepare for Download Phase
55	Requests in Fetch Download Phase
56	Requests in End Synchronization Phase
57	Requests in Send Download Phase
58	Requests in Wait for Download ACK Phase
59	Requests in Get DB Worker ACK phase
60	Requests in Connect for Download ACK Phase

61	Requests in Non-Blocking Download ACK Phase
-36	Synchronization Started
-37	Synchronization Finished
2	Pages Used
3	Pages Locked
5	TCP Connections Opened
6	TCP Connections Closed
7	TCP Connections Rejected
8	TCP Bytes Read
9	TCP Bytes Written
12	Pages Swapped Out
13	Pages Swapped In
22	Commits
23	Rollbacks
10	Connected Clients
24	Successful Synchronizations
25	Failed Synchronizations
26	Errors
27	Warnings
28	Database Connections in Use
33	Rows Downloaded
34	Rows Uploaded
68	RTNotifier Lightweight Polls
69	RTNotifier Lightweight Polls Hits
-39	RTNotifier Lightweight Polls Misses
-18	Connections Opened Rate
-19	Connections Closed Rate
29	Raw TCP Stage Length
30	Stream Stage Length
31	Heartbeat Stage Length
32	Command Processor Stage Length
65	Timed Work Stage Length
66	Notifier Stage Length
70	Outbound Enabler Stage Length
63	Server Tracked Memory Usage
15	Pages in Stream Stack
-13	Percent of Pages in Stream Stack