

SerialMagic BlackBerry API Developer's Guide

SerialMagic for BlackBerry is a powerful program that allows data integration between Bluetooth barcode scanners, Bluetooth RFID scanners, and other Bluetooth devices that send ASCII data and existing applications (i.e. applications do not need to be barcode scanner aware, or RFID scanner aware). In some cases it may be desired to for applications to integrate directly with the powerful SerialMagic engine.

While applications could be written directly to the target devices, doing so is tedious and burdensome. It also can require special code be developed for each device. Using SerialMagic BlackBerry Engine via the SerialMagic BlackBerry API allows very quick and painless support of devices that are currently supported by SerialMagic.

Using the SerialMagic BlackBerry API also allows applications to get data from SerialMagic when the BlackBerry Keystroke Injection policy is set to deny.

Using the SerialMagic BlackBerry API Scan Event Feature

In order to do this, follow these simple guidelines. This document assumes reader is familiar with BlackBerry developer concepts.

The GUID for the SerialMagic Professional global event is **0xDB34DB43DB119DB0**

Your application must implement the *GLOBALEVENTLISTENER* interface. This interface defines the method:

```
EVENTOCCURRED(LONG GUID, INT DATA0, INT DATA1, OBJECT OBJECT0, OBJECT OBJECT1)
```

An example implementation of this interface is as follows:

```
PUBLIC VOID EVENTOCCURRED(LONG GUID, INT DATA0, INT DATA1, OBJECT OB1, OBJECT OB2) {  
    IF (GUID == SERIALMAGICBB.GLOBAL_DATA_EVENT) {  
        STRING SCAN = OB1.TOSTRING();  
        SYSTEM.OUT.PRINTLN("RECEIVED GLOBAL SCAN EVENT WITH DATA: " + SCAN);  
    }  
}
```

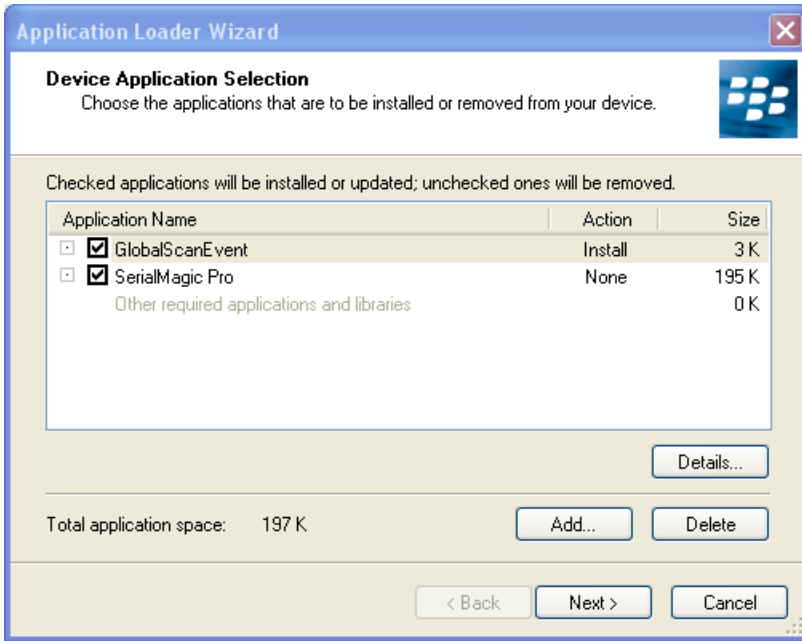
SerialMagic will supply the scan to this GlobalEvent as a *STRING* contained in *OBJECT OBJECT1*. The example above shows how to retrieve the scan. Please refer to the Java API documentation for further information about the GlobalEventListener interface.

You must register your application as a global event listener. This can be done with the call to *APPLICATION.ADDGLOBALEVENTLISTENER(GLOBALEVENTLISTENER G)*. Once you have registered your application, it will receive scan events from SerialMagic when the SerialMagic Action (SMA) "EVENT-SCAN" is included somewhere in your SerialMagic Action Profile (SMAP). SerialMagic will not automatically post the GlobalEvent if this SMA is not included in the SMAP.

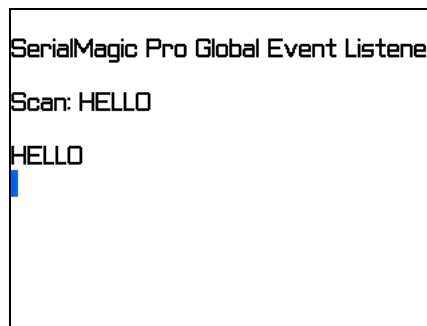
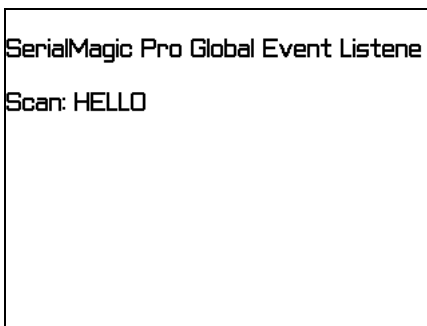
For a complete listing of Global SerialMagic events, see Appendix A.

SerialMagic Global Scan Even Demonstration Application - GlobalScanEvent

Serialio.com provides a simple application complete with source code that demonstrates how a BlackBerry application can receive a scan event from SerialMagic. The screen below demonstrates how the GlobalScanEvent application is loaded on a BlackBerry using the BlackBerry Desktop Manager.



To test, first Start SerialMagic. Insure the selected SMAP (SerialMagic Action Profile) has an EVENT-SCAN entry. Connect with the Bluetooth barcode or RFID scanner, and select it as the active device for SerialMagic (see SerialMagic documentation) then launch the GlobalScanEvent application. When a scan occurs it will be displayed as shown (below-left) in the GlobalScanEvent view. Note that if the SMAP also includes the DATA-IN, and ENTER Action entries in addition to the EVENT-SCAN Action, then the output will similar to below-right. The “Scan: HELLO” is from the event, and the other HELLO is from the keystroke injection input the GlobalScanEvent application.



Make SerialMagic Better

Please provide your feedback; Serialio.com is eager to continue to improve SerialMagic.

Appendix A – SerialMagic Events

This appendix describes how to use Global Events as published by SerialMagic Professional.

The following defines are the values that should be used for the Global Data Event, and Global SMA Event. Global Data Event only posts scan data, while Global SMA Event can post an assortment of different items.

```
public static final long GLOBAL_DATA_EVENT = 0xDB34DB43DB119DB0L;
public static final long GLOBAL_SMA_EVENT = 0xDB34DB43DB119DB3L;

public static final long GLOBAL_GMA_EVENT = 0xDB34DB43DB119DB4L;
public static final long GET_BT_DEVICES_GLOBAL_EVENT = 0x94d3fa4df3f34a8L;
public static final long SM_CONNECT_GLOBAL_EVENT = 0x1e79efe29b4d96d9L;
public static final long REGISTER_GLOBAL_EVENT = 0xc22363A3FC3E009FL;
public static final long REGISTER_STATUS_EVENT = 0xf7c183231827da51L;

public static final long SM_GET_SMAPS_GLOBAL_EVENT = 0x495110f3bcac4897L;
public static final long SM_SET_ACTIVE_SMAP_GLOBAL_EVENT = 0xf1fa77b66012a0f9L;
```

Below are descriptions of the different types of global events that can be published by SerialMagic Professional.

GLOBAL-DATA-EVENT

This event will simply post scan data, and nothing else. The ID for this event is GLOBAL_DATA_EVENT as defined above. The scan data will be formatted as a String and placed as Object object0 in the postGlobalEvent() method of the ApplicationManager class. For example, postGlobalEvent will be called as:

```
postGlobalEvent(GLOBAL_DATA_EVENT, 0, 0, data, null);
```

In this case, *data* is the String that contains the scan data. This will NOT be formatted as XML, but will only contain the scan data.

GLOBAL-SMA-EVENT

The Global SMA Event will be used for all other types of global events that are published by SerialMagic Professional. For these events, the ID used is GLOBAL_SMA_EVENT. Each of these sub-events formats the data similar to XML. All of these events will call postGlobalEvent() as follows:

```
postGlobalEvent(GLOBAL_SMA_EVENT, 0, 0, xml_string, null);
```

Here, *xml_string* is the important data which contains the information of this call. Below are defined all of the types of data that can be formatted for GLOBAL-SMA-EVENT.

Use the SMA <<EVENT-SCAN>> to configure SerialMagic to send the GLOBAL_SMA_EVENT.

SMA-EVENT-QTY (use SMA <<EVENT-QTY>>)

This event type is used to pass in quantity information for an associated scan event. The xml_string for this event will be formatted like:

```
<SMA_EVENT_QTY>5</SMA_EVENT_QTY>
```

Where '5' is the quantity of items.

SMA-EVENT-TAB (use SMA <<EVENT-TAB>>)

This event is useful if the user wants to tab over to the next column in a spreadsheet. It is formatted:

```
<SMA_EVENT_TAB/>
```

SMA-EVENT-LEFT (use SMA <<EVENT-LEFT>>)

This event is similar to the TAB event, but moves the cursor one cell to the left in the spreadsheet. It is formatted:

```
<SMA_EVENT_LEFT/>
```

SMA-EVENT-UP (use SMA <<EVENT-UP>>)

This event will move the cursor one cell up in the spreadsheet. It is formatted:

```
<SMA_EVENT_UP/>
```

SMA-EVENT-DOWN (use SMA <<EVENT-DOWN>>)

This event will move the cursor one cell down in the spreadsheet. The format is:

```
<SMA_EVENT_DOWN/>
```

SMA-EVENT-NEW-ROW (use SMA <<EVENT-NEW-ROW>>)

This event is used when the user wishes to move to the beginning of the following row and is formatted:

```
<SMA_EVENT_NEW_ROW/>
```

SMA-EVENT-FIND (use SMA <<EVENT-FIND>>)

This event will be used like a calling function. It will be useful when a user wishes to know if the scan data is already contained in the spreadsheet. If it is, the user may wish to update the quantity of this item, otherwise simply insert the scan. This event will be formatted (if the scan data is "0123456789"):

```
<SMA_EVENT_FIND>0123456789</SMA_EVENT_FIND>
```

Once this event is posted, SerialMagic Professional will wait for a response from GridMagic, which is formatted as:

<GMA_EVENT_FOUND>YES</GMA_EVENT_FOUND>

or

<GMA_EVENT_FOUND>NO</GMA_EVENT_FOUND>

to indicate whether or not the scan was found. The GMA_EVENT_FOUND event should use the ID GLOBAL_GMA_EVENT as defined above. An example of this event would be:

```
String found = "<GLOBAL_EVENT_FOUND>YES</GLOBAL_EVENT_FOUND>";  
postGlobalEvent(GLOBAL_GMA_EVENT, 0, 0, found, null);
```

Note that currently there is a 3 second timeout on the SMA_EVENT_FIND event. If a GMA_EVENT_FOUND event is not issued within this timeout period, it will be assumed that the data was not found.

SMA-EVENT-POST-ACK (use SMA <<EVENT-POST-ACK>>)

This event can be used to post data to an application using Blackberry message passing. It is the target application's responsibility to acknowledge this event by issuing a global event in response. The format of this event is as follows (if scan data is "123456789"):

<SMA_EVENT_POST_ACK>123456789</SMA_EVENT_POST_ACK>

The acknowledgement to this event should be formatted:

<GMA_EVENT_ACK>

for the data in this event while the ID for this event should be GLOBAL_GMA_EVENT. For example, a response could be formatted as follows:

```
String ack = "<GMA_EVENT_ACK>";  
postGlobalEvent(GLOBAL_GMA_EVENT, 0, 0, ack, null);
```

SMA-EVENT-FIRST-BLANK-ROW (use SMA <<EVENT-FIRST-BLANK-ROW>>)

This event will be used to instruct the GridMagic (or other SerialMagic compliant spreadsheet application) to search for the first blank row of the spreadsheet.

<SMA_EVENT_FIRST_BLANK_ROW/>

SMA-EVENT-NEXT-BLANK-ROW (use SMA <<EVENT-NEXT-BLANK-ROW>>)

This event will be used to instruct the GridMagic (or other SerialMagic compliant spreadsheet application) to search for the next blank row of the spreadsheet based on the current cursor location in the spreadsheet.

<SMA_EVENT_NEXT_BLANK_ROW/>

GET-BT-DEVICES-GLOBAL-EVENT

In order to receive the list of paired Bluetooth devices, third party application should send Get Bluetooth Devices Event.

```
postGlobalEvent(GET_BT_DEVICES_GLOBAL_EVENT, 0, 0, null, null)
```

SerialMagic Professional will reply by posting a global event with the same ID. *Object ob1* parameter contains String array with Bluetooth addresses. *Object ob2* contains String array with device friendly names list. Element indexes correspond between two arrays.

SerialMagic professional must be launched before issuing this command. To activate SerialMagic Professional autorun feature, expand SerialMagic Options branch and set “Start on device reset” to Yes. If SerialMagic Professional is not started, this event will not be replied.

SM-CONNECT-GLOBAL-EVENT

This event should be posted by a third-party application to establish or stop connection between SerialMagic Professional and one of the paired Bluetooth devices.

```
postGlobalEvent(SM_CONNECT_GLOBAL_EVENT, data0, data1, obj1, obj2)
```

int data0 – Bluetooth device type. Currently supported types are:

Device type	Parameter value
LaserChamp BT	0
CHS7	1
Other	2
MS9535	3
MS1633	4
SF51	5
KDC200	6
IDChamp	7
Restock SS BT-M2	8
Scanfob 2002	9

int data1 determines whether to start or stop Bluetooth connection.

```
int API_CONNECT_CMD      = 0 – used to connect to the selected device  
int API_DISCONNECT_CMD  = 1 – used to disconnect from the device
```

Object obj1 – Bluetooth address to connect to.

Object obj2 – not used.

SerialMagic Professional will initiate connection and post global event with the same ID. *int data0* value will demonstrate the result of operation.

```
int API_CONNECT_SUCCESS      = 0 – SerialMagic successfully connected to the selected device
int API_BLUETOOTH_DISABLED  = 2 – Bluetooth is disabled on the BlackBerry device
int API_CONNECT_FAILED      = 1 – reserved for future needs
```

REGISTER-GLOBAL-EVENT

Register Event can be used to register SerialMagic Professional via registration web service. To use this feature, third party application should fire global event with the ID REGISTER_GLOBAL_EVENT. The required parameter Username should be passed through *Object object0* argument of the **ApplicationManager.postGlobalEvent(long guid, int data0, int data1, Object object0, Object object1)** method. Note that SerialMagic Professional must be connected to the supported barcode scanner when attempting to register via the web service.

SerialMagic Professional will post global event with the same ID to notify third party application of the result of registration. *Object object0* parameter will contain String object with the received serial number if the operation was successful or the detailed error message if the operation failed.

SM-GET-SMAPS-GLOBAL-EVENT

This event is used to get a list of SerialMagic Profiles aka SMAPS (SerialMagic Action Profiles) This event is posted by the 3rd party app. SerialMagic Professional will post global event with the same ID. *Object ob1* value will contain the list of existing SMAP names. Class cast to *String[]* is required

SM-SET-ACTIVE-SMAP-GLOBAL-EVENT

This event is used to set the active SerialMagic Profile aka SMAP (SerialMagic Action Profile) This event is posted by the 3rd party app. *Object ob1* should contain the **name** of the existing profile. In case of invalid profile name this command is ignored.