ZEBRA

MDM TOOLKIT

ANDREW KNIGHT

PRODUCT MANAGER

THE ENTERPRISE CHALLENGE

ENTERPRISES NEED TO KEEP PACE WITH CONSTANT CHURNING OF:

1. ENTERPRISE DEVICES
2. CONSUMER DEVICES
3. OPERATING SYSTEMS / APPS
4. SECURITY REQUIREMENTS

WHAT IS THE MDM TOOLKIT?

- **Documentation, Sample Code, Utilities & Packages for integrating MX management capabilities into EMM / MDM Solutions**
- Component of Zebra's EMM/MDM Product Strategy
- Enables Enterprise Device Management capabilities for all Third-Party MDM Vendors
- Think of as "EMDK for MDM Developers", but is not EMDK 
- Built on  , targets MX version 4.4 feature set

EMM / MDM ECOSYSTEM

App Developers

EMM/MDM Developers

Device Administrators

SOTI

airwatch by vmware

wavelink

Windows Intune

CITRIX XenMobile

MobileIron

SAP Afaria

mprodigy

WHAT YOU NEED TO KNOW

- It IS for developers of EMM/MDM solutions only
- It IS leveraging the same underlying Zebra value-added MX features as EMDK/StageNow
- Remote Control is a requirement for many MDM solutions

COMMON MISCONCEPTIONS:

- It is not a new EMDK or a new architecture layer on top of MX
- It is not ONLY a documentation deliverable of MX
- It is not a client signing procedure
- It is not intended for anyone other than third-party EMM/MDM solution development teams

KEY DIFFERNECES

- **TARGET AUDIENCE – only MDM Developers, EMDK is for general app developers; they have different needs**
- EMDK fully helpdesk supported, Toolkit has targeted vendor support only
- **Fundamental Use Case Differences**
  - EMDK has a runtime (helps developer), Toolkit does not (submits directly to MXMS framework)
  - EMDK uses mostly canned XML, occasional dynamic XML; Toolkit uses mostly dynamic XML
    - EMDK - Special key mapping in application, use profile manager (hard code values, static XML)
    - TK – Clock, set time always needs to use dynamic parameters

What does it mean for me?

- MDM Developers – support for managing Zebra devices
- Device Administrators – more choices for MDMs
- Biz App Developers – more MDMs, greater chance that my app will be deployed/managed by MDMs
  - You are leveraging MX... And so is MDM vendor (may be coexisting or conflicting with same function)
  - Everybody uses MX (TK, EMDK, StageNow) – might need to work with administrator to determine best way to configure (ie. Key Mapping)
  - Consider how your app should behave in a managed environment?

Enterprise Mobility Management (EMM) at a Glance

1 Deployment:
- Out-of-Box Staging / Self Enrollment
- Provisioning & Device Inventory
- Set security & user policies

2 App Management:
- Deploy/Update/Configure
- Silent Install & Un-install
- Whitelist/Blacklist

3 Security & Content Mgmt:
- File Management / Versioning
- Content Filter & Secure browser
- Containerization & Cert Mgmt
- SharePoint & AD Single Sign-On

4 Device Management:
- Asset Tracking & Reporting
- Inventory & Content Updates
- User & Location-based policies
- OTA OS & Patch Updates

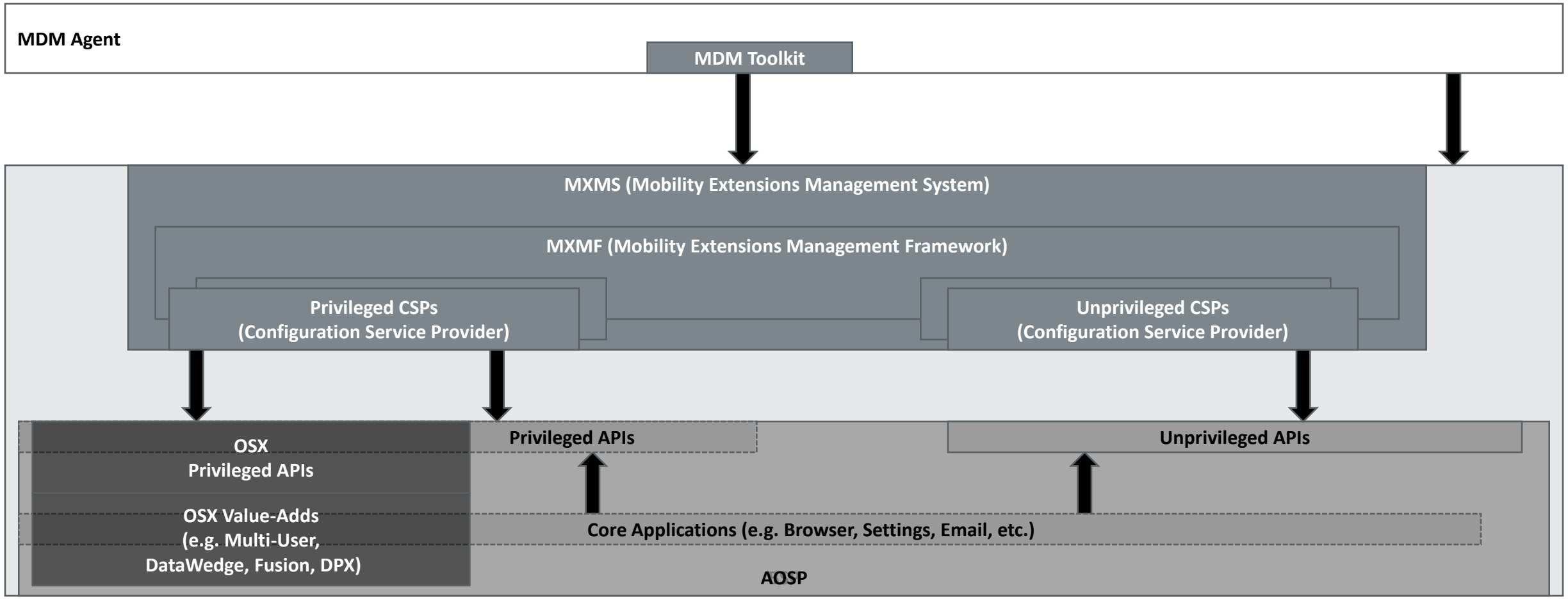
5 Troubleshooting:
- Device Diagnostics
- Help Desk & Remote Control
- Remote Lock & Wipe

6 Decommissioning:
- Compliance Tracking
- Backup Data
- Locate Device & Un-enroll

Features – MX 4.4

Device Configuration	UI Settings Control	Security	Application Management	Wireless
Select date and time format	Control Mute/Vibrate	Control Ethernet	Close all settings activities	Set DHCP options
Wipe SD card	Keyboard/Button remapping	NFC On/Off	Force application to stop	Control background data
Configure Analytics (transfer mode)	Control access to App Info (Settings)	Enable/Disable Imager	Enable/disable browser address bar	Control data roaming
Battery decommission thresholds	Lockdown Unknown app setting	Enable/Disable Rear Camera	App White Signature Verify	WAN On/Off
Set device host name	Disable Access Wifi Settings	Enable Disable Front Camera	App Source Ctrl (unknown)	WLAN Profile(s) Settings
Screen Timeout Set	Button Remapping	SD Card Use Enable/Disable	Recent App Use Listing	WLAN Sleep Policy
XML Process Controls	Keyboard remapping	Screen Lock Timeout	Mulit-User App Enable/Disable	On/Off SSID Notification
Control XML Persistence	Auto-Correct Enable/Disable	Device Admin Controls	Persist App in Multi-user	APN Configuration
Touch Sensitivity Ctrl	Set localizations	Encrypt/Unencrypt SD Card	Set Default Launcher	
Invoke Factory Reset	Disable Setting-Ent Reset	Encrypted XML Controls	Start App or Service	
Invoke Ent Reset		Control Bluetooth	Silent App Upgrade	
NTP Config		USB-Disable ext drive connect	Silent App Install/Remove	
Time Zone On/Off		Selective USB (MTP, PTP)		
Auto Time Config		USB - Control Dev as Storage		
OS Update		Control USB ADB Only		
		Control USB Mode		
		Control Clipboard		
		App XML Enablement		
		Initialize Key Store		
		Initiate Device Wipe		
		Cert Auto Clock Set		

MDM Agent Interface to MXMS



MDM Agent
Interface to MXMS

- Since Tier 2 and Tier 3 MDM Agents are **not** signed, they MUST perform ALL privileged management functions by interfacing to MXMS
- This interface is documented and enabled by the Toolkit for MDMs and consists of:
  - Constructing a suitable Request XML Document
  - Submitting the Request XML Document to MXMS
  - Receiving a Result XML Document back from MXMS
  - Parsing the Result XML Document to determine result

Quick Start

Overview

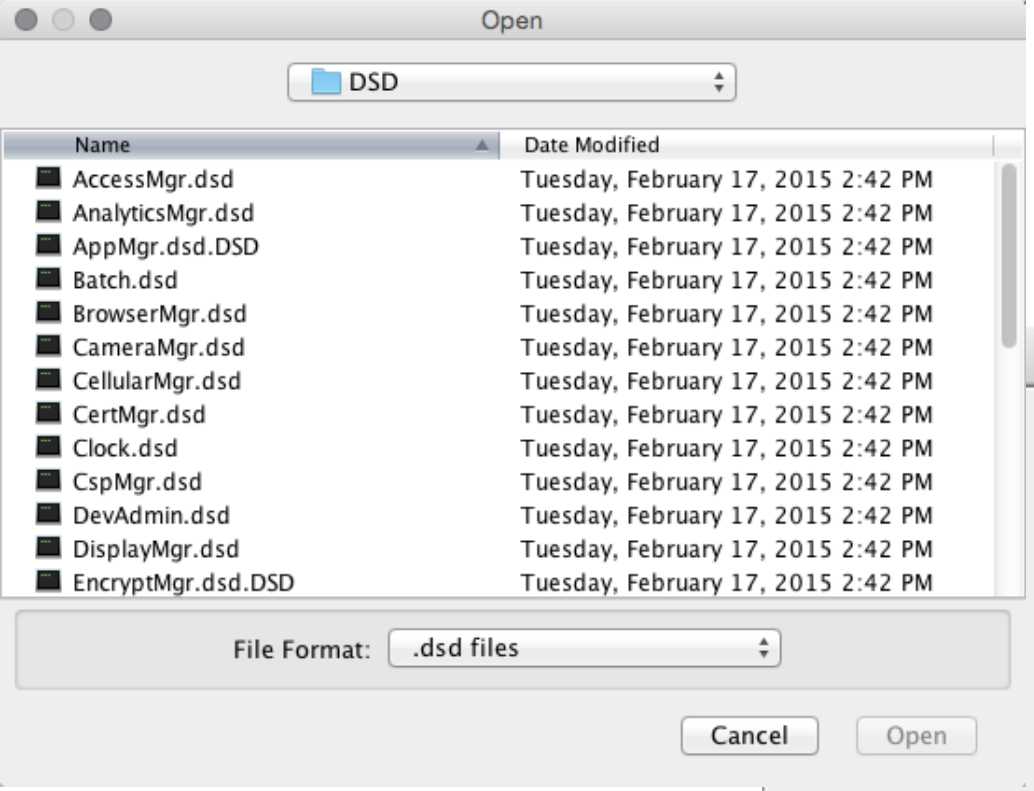
This quick start guide will walk through the common tasks and components that you will use in order for your MDM client to interface with the MX Management System available on Zebra Android devices. The following steps will be covered.

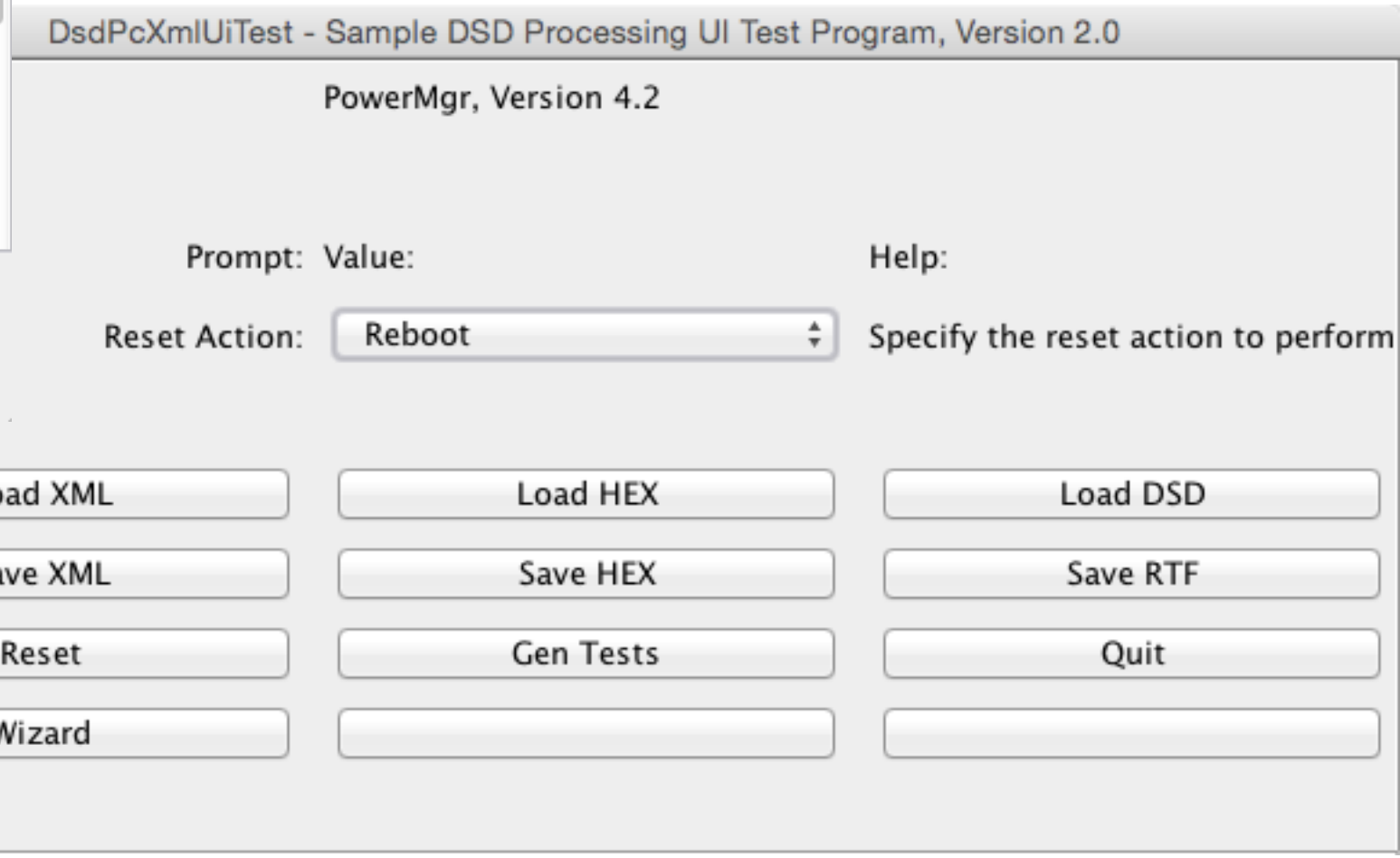
- **Intro To The MXMS** - The basic information that is needed for using the MX Management System.
- **Binding to the MXMS** - All communications to the MXMS on Zebra devices, occur through a common binding interface.
- **Generating XML** - Data exchanged to the MXMS from the MDM client is handled through a defined XML structure. Using the DSDtoXml tool provided in the MDM Toolkit will provide a template for the XML to be used for applying settings. Typically you will use this tool to generate all needed functions and then replace string values for dynamically changing variables.
- **Submitting XML** - Within the MDM client, XML will be submitted to apply settings via a simple API.
- **Querying the MXMS** - Within the MDM client, XML will be submitted to query the MXMS to receive back information about the current settings on the device.
- Next Steps

Requirements

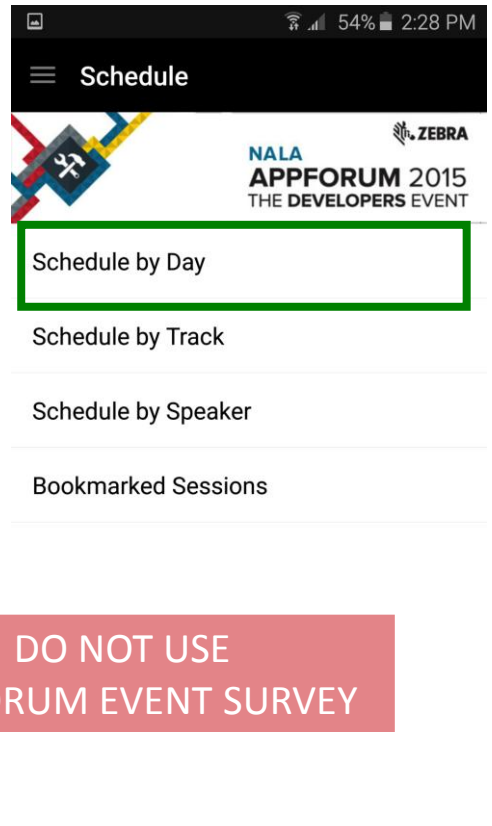
- MDM Toolkit (XML Generator Tool, DSD Files)
- Symbol Android Device with MX
- Java JVM Installed
- Android ADT

Intro To The MXMS

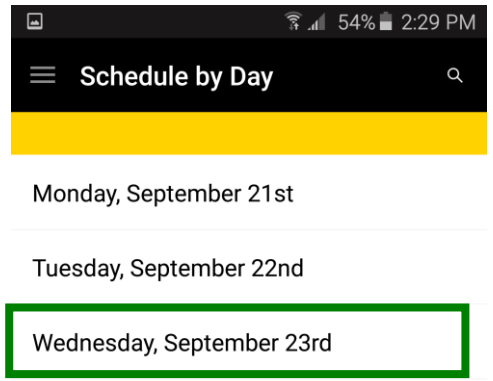

Open
DSD
Name
Date Modified
AccessMgr.dsd Tuesday, February 17, 2015 2:42 PM
AnalyticsMgr.dsd Tuesday, February 17, 2015 2:42 PM
AppMgr.dsd.DSD Tuesday, February 17, 2015 2:42 PM
Batch.dsd Tuesday, February 17, 2015 2:42 PM
BrowserMgr.dsd Tuesday, February 17, 2015 2:42 PM
CameraMgr.dsd Tuesday, February 17, 2015 2:42 PM
CellularMgr.dsd Tuesday, February 17, 2015 2:42 PM
CertMgr.dsd Tuesday, February 17, 2015 2:42 PM
Clock.dsd Tuesday, February 17, 2015 2:42 PM
CspMgr.dsd Tuesday, February 17, 2015 2:42 PM
DevAdmin.dsd Tuesday, February 17, 2015 2:42 PM
DisplayMgr.dsd Tuesday, February 17, 2015 2:42 PM
EncryptMgr.dsd.DSD Tuesday, February 17, 2015 2:42 PM
File Format: .dsd files
Cancel
Open

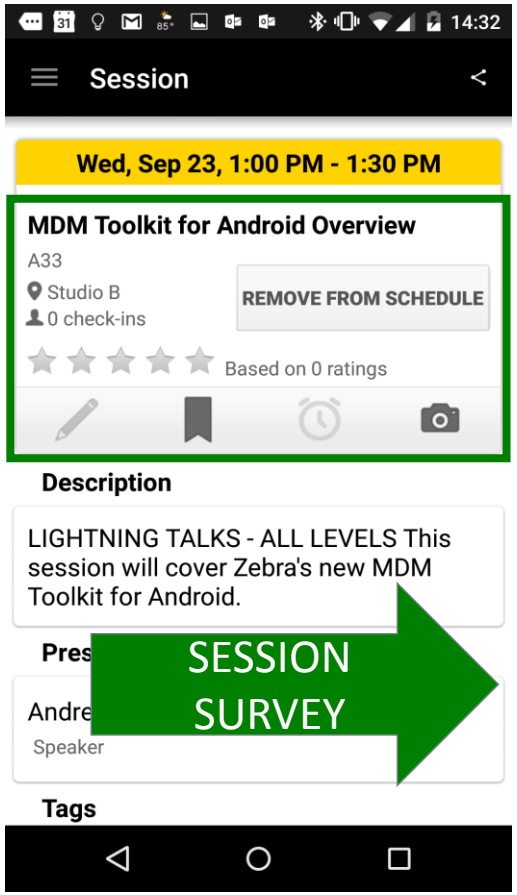

DsdPcXmlUiTest - Sample DSD Processing UI Test Program, Version 2.0
PowerMgr, Version 4.2
Prompt: Value:
Help:
Reset Action: Reboot
Specify the reset action to perform
Load XML
Load HEX
Load DSD
Save XML
Save HEX
Save RTF
Reset
Gen Tests
Quit
Wizard

PLEASE TAKE THE SURVEY

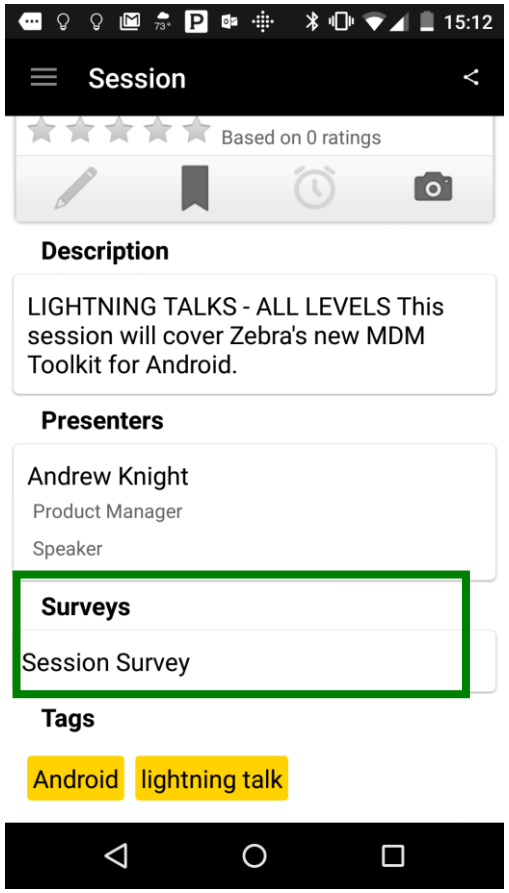


DO NOT USE
APPFORUM EVENT SURVEY





SESSION
SURVEY



ZEBRA

THANK YOU