UEFI Manageability and REST Services

Fall 2017 UEFI Seminar and Plugfest
October 30 – November 3, 2017
Presented by Abner Chang (HPE), Ting Ye (Intel)
Agenda

- Introduction
- UEFI Manageability
- Real Use Case: UEFI iSCSI
- REST in UEFI
- Insufficient Capabilities in ‘EFI REST’ and Proposals for Enhancement
- Summary & Call to Action
Introduction
Introduction

• Today’s platforms have amazing capabilities, but they are complex to configure
  – Firmware configuration model is still focused on “Press <DEL> to Enter Setup”
  – Remote management isn’t consistent

• UEFI has abundant firmware manageability interfaces, and is adopting REST/Redfish, but there are still gaps in the standard
Goals

• Overview of UEFI manageability interfaces & Human Interface Infrastructure (HII)
• Examples of handling configuration data
• Overview of EFI REST protocol
• Proposals to cover gaps in REST services
• Background of REST service provider and UEFI REST Client
UEFI Manageability
UEFI Manageability

• Human Interface Infrastructure
  — HII Protocols
  — x-UEFI language
• Keyword Handler Protocol
• Configuration Namespace
Human Interface Infrastructure

- HII Database
  - Package containing forms + strings
- Add-in Device
- UEFI Driver
- Export Data
- O/S Boot
  - Forms Browser/Processor

UEFI Plugfest – October 2017
www.uefi.org
HII Protocols

• Content Registration:
  – HII Font, Font Ex, Font Glyph Generator
  – HII Image, Image Ex Image Decoder
  – HII String
  – HII Database

• Browser Protocol:
  – Config Access / Routing
  – Form Browser
  – HII Popup
Multiple Language Support

Typical Data Contained in HII Database

UEFI Plugfest – October 2017
www.uefi.org
**x-UEFI: a “UEFI” Language**

- “machine-to-machine” language

<table>
<thead>
<tr>
<th>String Pack Language</th>
<th>ENG</th>
</tr>
</thead>
<tbody>
<tr>
<td>Token #1</td>
<td>“BIOS Vendor”</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>String Pack Language</th>
<th>SPA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Token #1</td>
<td>“Vendedor de BIOS”</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>String Pack Language</th>
<th>x-UEFI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Token #1</td>
<td>“BIOS_VENDOR_KEYWORD”</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>String Pack Language</th>
<th>NULL</th>
</tr>
</thead>
</table>
How To Use This Language

- Retrieve platform exported data
- Get String Token of x-UEFI keyword
- Find op-code using the string token

<table>
<thead>
<tr>
<th>Offset 0</th>
<th>Offset 4</th>
<th>Offset 8</th>
<th>Offset 12</th>
</tr>
</thead>
<tbody>
<tr>
<td>Op-Code</td>
<td>Length</td>
<td>Prompt Token #</td>
<td>Help Token #</td>
</tr>
<tr>
<td>Question ID</td>
<td>VarStore ID</td>
<td>Flags</td>
<td>Op-Code Specific</td>
</tr>
<tr>
<td>VarStoreInfo</td>
<td>Op-Code Specific</td>
<td>Op-Code Specific</td>
<td>Op-Code Specific</td>
</tr>
</tbody>
</table>
Real Use Case: UEFI iSCSI
Real Use Case: UEFI iSCSI

• iSCSI is a popular network boot target

• UEFI iSCSI defines a number of dynamic HII forms
iSCSI x-UEFI keywords

- Registered to
  [http://uefi.org/confignamespace](http://uefi.org/confignamespace)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>ReadWrite/ReadOnly</th>
<th>ReadWrite/ReadOnly</th>
</tr>
</thead>
<tbody>
<tr>
<td>iSCSIInitiatorName</td>
<td>ReadWrite</td>
<td>iSCSIMacAddr</td>
</tr>
<tr>
<td>iSCSIAttemptName:</td>
<td>ReadOnly</td>
<td>iSCSIAddAttempts</td>
</tr>
<tr>
<td>iSCSIBootEnable:</td>
<td>ReadWrite</td>
<td>iSCSIDeleteAttempts</td>
</tr>
<tr>
<td>iSCSIConnectRetry:</td>
<td>ReadWrite</td>
<td>iSCSIDisplayAttemptList</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

UEFI Plugfest – October 2017  www.uefi.org
x-UEFI Usage Example

iSCSIInitiatorName

VFR file

```plaintext
string varid = ISCSI_CONFIG_IFR_NVDATA.InitiatorName,
   prompt = STRING_TOKEN(STR_ISCSI_CONFIG_INIT_NAME),
```

UNI file

```plaintext
#string STR_ISCSI_CONFIG_INIT_NAME   #language en-US "iSCSI Initiator Name"
#string STR_ISCSI_CONFIG_INIT_NAME   #language x-UEFI "iSCSIInitiatorName"
```

Script file

```plaintext
IScsiScript -i iqn.edkii.intel.com
```
General Guidelines

- **OEMs ...**
  - Get keywords definition from http://uefi.org/confignamespace
  - UseKeywordHandler.GetData/SetData

- **Firmware vendors ...**
  - Get HII updates from the latest Intel® UEFI Development Kit

- **IHVs ...**
  - Define and register x-UEFI keywords
  - Support keyword setting in ConfigAccess.RouteConfig
REST in UEFI
REST in UEFI

**REST = REpresentational State Transfer (architecture)**

Most common client/server REST interface is HTTP

HTTP protocol

HTTP verbs like PUT, GET, POST, PATCH, HEAD and DELETE

URL

XML, JSON and etc.

HTTP request header and body.

HTTP response header and body.
How does REST Relate to UEFI?

- Setup Browser
- HII Forms
- Config Access / UEFI Variable
- HII Database
- Device
- UEFI Driver

UEFI Plugfest – October 2017

www.uefi.org
How does REST Relate to UEFI?

Platform Management Driver

Keyword Handler Protocol

Setup Browser HII Forms

Config Access / UEFI Variable

UEFI Driver

#x-UEFI Keyword "iSCSIInitiatorName"

EFI_REST_PROTOCOL

Internet

Redfish Resources

Configure Redfish BIOS property Remotely
How does REST Relate to UEFI?

Platform Management Driver

Keyword Handler Protocol

Setup Browser HII Forms

Internet

EFI_REST_PROTOCOL

Redfish Resources

UEFI Driver

UEFI Variable

EFI_REST_PROTOCOL

#x-UEFI Keyword "iSCSIInitiatorName"

Configure Redfish BIOS property Remotely

BIOS Redfish Attribute "iSCSIInitiatorName"
How does REST Relate to UEFI?

- Platform Management Driver
- Redfish Resources
- Redfish BIOS property Remotely
- BIOS Redfish Attribute "iSCSIInitiatorName"

#x-UEFI Keyword "iSCSIInitiatorName"

Configure Redfish BIOS property Remotely
How does REST Relate to UEFI?

Platform Management Driver

Keyword Handler Protocol

Setup Browser HII Forms

HII Database

Device

UEFI Driver

Redfish Resources

BMC

#x-UEFI Keyword "iSCSIInitiatorName"

Configure Redfish BIOS property Remotely

BIOS Redfish Attribute "iSCSIInitiatorName"
How does REST Relate to UEFI?

BIOS Redfish Attribute "iSCSIInitiatorName"
How does REST Relate to UEFI?

- Platform Management Driver
- Keyword Handler Protocol
- HII Form
- HII Database
- UEFI Driver

Client

Resources

EFI_REST_PROTOCOL

Server

Resource

#x-UEFI Keyword "iSCSIInitiatorName"

Configure Redfish BIOS property Remotely

BIOS Redfish Attribute "iSCSIInitiatorName"
EFI REST Protocol in UEFI Spec v2.5

EFI REST Protocol use HTTP-like message as the format of REST service request and response

typedef struct _EFI_REST_PROTOCOL {
  EFI_REST_SEND_RECEIVE SendReceive;
  EFI_REST_GET_TIME GetServiceTime;
} EFI_REST_PROTOCOL;
EFI REST Protocol in UEFI Spec v2.5

EFI REST driver instances may use different underlying protocol to communicate with REST server.

#x-UEFI Keyword "iSCSIInitiatorName"

BIOS Redfish Attribute "iSCSIInitiatorName"

Client

Request

Server

Resource

Through EFI Network stack

Platform specific protocol to REST server provided by BMC
Insufficient Capabilities in ‘EFI REST’ and Proposals for Enhancement
Are ‘EFI REST’ Capabilities Sufficient?

• Are ‘EFI REST’ capabilities sufficient for UEFI Manageability?
• Are ‘EFI REST’ capabilities sufficient for the modern REST Services?
Are ‘EFI REST’ Capabilities Sufficient?

How can we support a variety of in-band/out of band UEFI management models as shown in this figure?
Are ‘EFI REST’ Capabilities Sufficient?

How can we support On-platform/Off-Platform REST services for EFI REST clients?

- **EFI REST Client Driver**
- **EFI REST Client (Platform Management, HII configuration)**
- **EFI REST Client Application**

Diagram:
- **Client Request** to **Server Resource**
- **On-platform REST Service** (such as BMC or others)
- **Off-platform REST service**
  - Vendor-specific REST Service
  - **iSCSIInitiatorName**
Suggestions for EFI REST Protocol Enhancements

• Multiple EFI REST driver instances for different REST services
• The information of REST service
• The location of REST service
• Asynchronous request and response
• Address the events from REST service
Suggestions for EFI REST Protocol Enhancements

Multiple EFI REST driver instances for different REST services

Multiple EFI REST driver instances can be installed on system for communicating to different REST services. The EFI REST client drivers and applications can use the specific EFI REST protocol to access to REST service.
Suggestions for EFI REST Protocol Enhancements

The information of REST service?

Each EFI REST driver instance has to provide the information about REST service it supports.
Suggestions for EFI REST Protocol Enhancements

The location of REST service?

Each EFI REST driver instance has to provide the information about the location of REST service.
Suggestions for EFI REST Protocol Enhancements

Synchronous request and response leads to a downtime

SendReceive() function provided in EFI REST protocol is synchronous transfer. EFI REST client has to wait for the response once it sends the request out.
Suggestions for EFI REST Protocol Enhancements

Asynchronous request and response

Support asynchronous SendReceive() function to provide efficient REST request and response. Also provides the better user experience when users use EFI REST client application.

#x-UEFI Keyword "iSCSIInitiatorName"
Suggestions for EFI REST Protocol Enhancements

Address the events from REST service

Most of REST service support event capability to indicate the changes of resource. User can register the event for certain type resource change such as resource created, modified, deleted and etc.
Summary & Call to Action
Summary

• UEFI provides interfaces for improved manageability
  — (HII, x-UEFI, Keyword handle protocol, EFI REST protocol)

• Improvements to REST are required for adoption of standardized manageability
Call to Action

• Define and register more x-UEFI keywords
• Enhance current EFI REST Protocol for flexible, scalable and user friendly interfaces for EFI REST Clients
  – Asynchronous request/response in the enhanced EFI REST Protocol
  – Event subscription for monitoring resource changes in enhanced EFI REST Protocol
  – Provide sufficient information for target REST service
Thanks for attending the Fall 2017 UEFI Plugfest

For more information on the UEFI Forum and UEFI Specifications, visit
http://www.uefi.org

presented by

Hewlett Packard Enterprise

Intel
Backup
What is REST?

**Representational State Transfer**, which is a software architecture style. There are some constraints applied to this architecture.

- **Uniform Interface**
  - Between client and server
  - Nouns and Verbs

- **Layered system**
  - SW, HW or middleware

- **Request**
  - Client and Server, separating the user interface from resource. Which improves portability and scalability.
  - *The request contains the unique identifier of resource and the request body contains the state (or state change) of that resource.*
  - *The state after server processed the request is returned in response body.*

- **Resource**
  - Resource could be identified
  - Separate resource from representation
  - Resource represented separately in certain formats
  - Resource manipulation through representations

**No client context stored in server. The necessary state is contained within request itself.**

*Client don’t have to concern the data storage*

*Server don’t have to concern the user interface and user state.*
REST in HTTP

HTTP is not the only interface (protocol) for REST architecture. However, it’s most commonly used in Web REST service. Any transport interface which is unified to REST service and have the well defined verb to manipulate resource could be the interface between REST server and REST client.

HTTP protocol
HTTP verbs like PUT, GET, POST, PATCH, HEAD and DELETE

HTTP request header and body.
HTTP response header and body.

Client
Request

Server
Resource

UEFI Plugfest – October 2017
www.uefi.org
REST Client and REST Services
REST Services

There are many REST Services and Providers. Most of those services provide RESTful APIs which are Web service API adhere to REST constraints. Those are called RESTful services.
REST Service Server and EFI REST Client

EFI REST Client Driver

EFI REST Client (Platform Management, HII configuration)

EFI REST Client Application

EFI REST Driver Instance

Server Resource

Server Resource

Server Resource

Server Resource

Vendor-specific REST Service

#x-UEFI Keyword "iSCSIInitiatorName"

iSCSIInitiatorName
Suggestions for EFI REST Protocol Enhancements

The location of REST service provided by specific EFI REST driver instance

Each EFI REST driver instance has to provide the information about the location of REST service it supports.
Suggestions for EFI REST Protocol Enhancements

Asynchronous request and response

SendReceive() function provided in EFI REST protocol is synchronous transfer. EFI REST client has to wait for the response once it sends the request out.

Support asynchronous SendReceive() function to provide efficient REST request and response. Also provide the better user experience when users use EFI REST client application.

Usage example, Add Prefer: respond-async to HTTP message header

Return HTTP 202 and URI for the status polling

Check URI status for the response

Return HTTP 200 to indicate the request is still in process.

Return HTTP 200 to indicate the response is ready.

Event Token

Signal event token to indicate the response is ready.

#x-UEFI Keyword "iSCSIInitiatorName"

UEFI Plugfest – October 2017
Suggestions for EFI REST Protocol Enhancements

Asynchronous request and response

The way how REST service returns final response to REST Protocol driver instance is REST service implementation-specific and transparent to REST client. The content of URI which pointed by HTTP Location header is REST service implementation-specific and not defined in REST Protocol specification. REST Protocol driver instance provider should have knowledge about how to poll the status of returning resource from given HTTP Location header.
Suggestions for EFI REST Protocol Enhancements

Address the events from REST service

Most of REST service support event capability to indicate the changes of resource. The user can register the event for certain type resource change such as resource created, modified, deleted, etc.

Enhancement of EFI REST Protocol

Event Service

Server Resource

Subscribe Event
Event sent to specific URI

Server Resource

Client Request

Most of REST service support event capability to indicate the changes of resource. The user can register the event for certain type resource change such as resource created, modified, deleted, etc.

#x-UEFI Keyword "iSCSIInitiatorName"

UEFI Plugfest – October 2017
Suggestions for EFI REST Protocol Enhancements

Multiple EFI REST clients to access to different REST service

Each EFI REST child instance communicates to different REST services.
Suggestions for EFI REST Protocol Enhancements

Multiple EFI REST clients to access to one REST service

Each EFI REST child instance communicates to different REST services. Or multiple EFI REST child instances access to the same REST service.

Enhancement of EFI REST Protocol