Order Management REST SPECIFICATION

Document Number: <###>

Document Version: : <V#.#>

Date: mmm, yyyy

Document Status: Draft

<Document Title>

NOTICE

Direct inquiries to the TM Forum office:

240 Headquarters Plaza,
East Tower – 10th Floor,
Morristown, NJ 07960 USA
Tel No. +1 973 944 5100
Fax No. +1 973 944 5110
TM Forum Web Page: www.tmforum.org
TM Forum Web Page: www.tmforum.org

<Document Title>

TABLE OF CONTENTS

<Document Title>

LIST OF TABLES

No table of figures entries found.

<Document Title>

INTRODUCTION

This document contains the REST API definition for Order Management.

Assumptions :

- OM system has access to the catalogue system

NB : Currently, this Order API contains some limitations :

- It only covers acquisition of products. i.e. new orders for new products only
- It does not manage bundling of orders
- The order begins to run directly upon creation
- Order modification or canceling in-flight is not possible

<Document Title>

RESOURCE MODEL

Managed Entity and Task Resource Models

<Document #, Version #>

<Document Title>

```
{"productOrder": {
  "id": "http://serverlocation:port/orderManagement/productOrder/42",
  "externalID": "NiceNameForTheConsumer_42",
  "description": "A wonderful 42 order for brand new products",
  "type": "residential",
  "state": "OPEN.RUNNING",
  "orderDate": "2013-07-23 08:16:39ZGMT+1",
  "completionDate": "2013-07-24 01:00:00ZGMT+1",
  "requestedCompletionDate": "2013-07-24 08:00:00ZGMT+1",
  "relatedParties": [
    {
      "role": "customer",
      "reference": "MSISDN-0033689770600"
    },
    {
      "role": "partner",
      "reference": " http://serverlocation:port/partnerManagement/partner/42"
    }
  ],
  "orderItems": [
    {
      "id": "1",
      "action": "add",
      "state": "NOT_STARTED",
      "productOffering": {"id": "http://serverlocation:port/catalogManagement/productOffering/42"},
      "product": {"productCharacteristics": [
        {
          "name": "Colour",
          "value": "White"
        },
        {
          "name": "Memory",
          "value": "16"
        }
      ]}
    },
    {
      "id": "2",
      "action": "add",
      "state": "NOT_STARTED",
      "productOffering": {"id": "http://serverlocation:port/catalogManagement/productOffering/43"},
      "product": {"productCharacteristics": [{
        "name": "anotherCharacteristic",
        "value": "itsValue"
      }]}
    }
  ]
```

<Document Title>

}}

Order states :
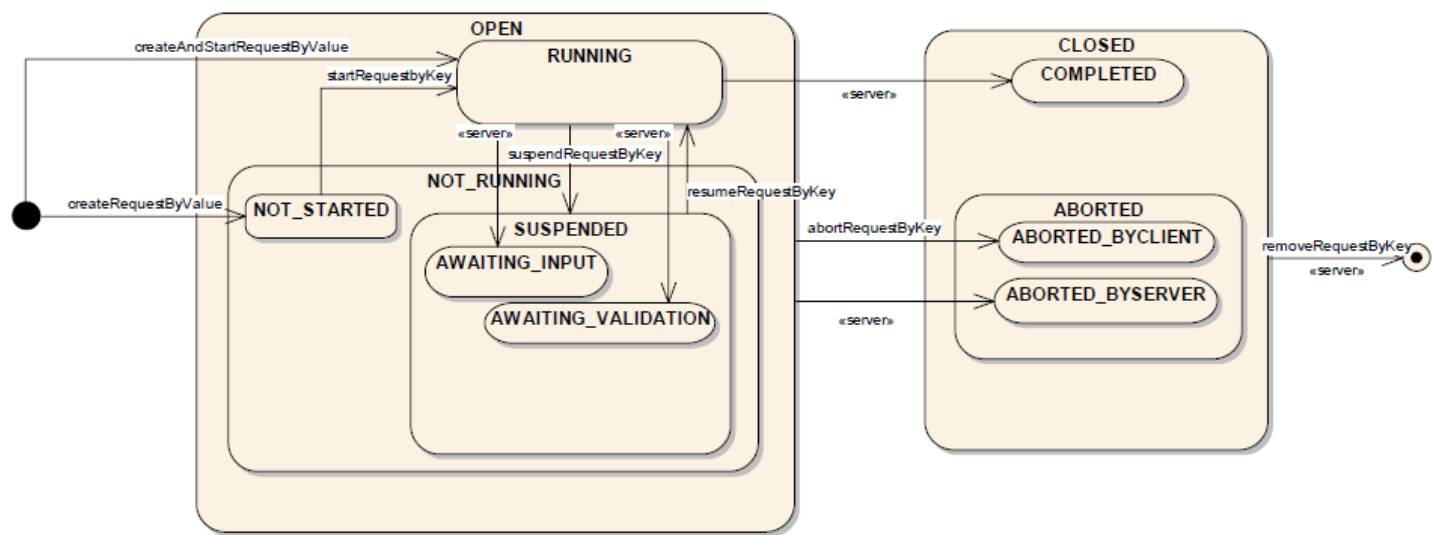
⇨ State descriptions

Only OPEN.RUNNING, CLOSED.COMPLETED and
CLOSED.ABORTED.ABORTED_BYSERVER are currently supported.

State	Description
OPEN.RUNNING	The order has been created and started
OPEN.NOT_RUNNING.NOT_STARTED	The order has been created but is not started yet
OPEN.NOT_RUNNING.SUSPENDED.AWAITING_INPUT	The order has been suspended because additional information is needed
OPEN.NOT_RUNNING.SUSPENDED.AWAITING_VALIDATION	The order has been suspended because it needs a validation from a party
CLOSED.COMPLETED	The order was successfully completed
CLOSED. ABORTED.ABORTED_BYCLIENT	The order was aborted as a result of a client request
CLOSED.ABORTED. ABORTED_BYSERVER	The order was aborted due to an internal error in the

<Document Title>

	provider system

Field Description :

- "Id" is the ID created on repository side (OM system)
- "externalID" is an ID given by the consumer and only understandable by him (to facilitate his searches afterwards)
- "type" is used to categorize the order from a business perspective that can be useful for the OM system (e.g. "enterprise", "residential", ...)
- "orderDate" is the date when the order was created
- "completionDate" is the date when the order was completed
- "requestedCompletionDate" is the expected delivery date
- "state" is described in the state-machine diagram above (currently limited to "OPEN.RUNNING", "CLOSED.COMPLETED", CLOSED.ABORTED.ABORTED_BYSERVER")
- "relatedParties"describe all the involved parties
- "orderItems" are the order lines :
  - "id" uniquely identifies the item within the order
  - "action" can be "add" / "modify" / "delete". (currently only "add" is supported)
  - "state" can be "NOT_STARTED", "RUNNING", "FAILED", "COMPLETED"
  - "productOffering" is the ordered offering (pricing, default values, etc. are fetched by the OM directly from the catalogue)
  - "product" is used to configure the product characteristics (only configurable characteristics and necessary only if a non default value is selected)

Notification Resource Models

Four notification types are defined :

- orderCreationNotification
- orderRemoveNotification (currently not supported)
- orderStateChangeNotification
- orderValueChangeNotification (currently not supported)

OrderCreationNotification

```
{"event": {"productOrder": {
    "id": "http://serverlocation:port/orderManagement/productOrder/42",
    "externalID": "NiceNameForTheConsumer_42",
    "description": "A wonderful 42 order for brand new products",
```

<Document Title>

```
      "type": "residential",
      "state": "OPEN.RUNNING",
      "orderDate": "2013-07-23 08:16:39ZGMT+1",
      "completionDate": "2013-07-24 01:00:00ZGMT+1",
      "requestedCompletionDate": "2013-07-24 08:00:00ZGMT+1",
      "relatedParties": [
         {
            "role": "customer",
            "reference": "MSISDN-0033689770600"
         },
         {
            "role": "partner",
            "reference": " http://serverlocation:port/partnerManagement/partner/42"
         }
      ],
      "orderItems": [
         {
            "id": "1",
            "action": "add",
            "state": "NOT_STARTED",
            "productOffering": {"id":
"http://serverlocation:port/catalogManagement/productOffering/42"},
            "product": {"productCharacteristics": [
               {
                  "name": "Colour",
                  "value": "White"
               },
               {
                  "name": "Memory",
                  "value": "16"
               }
            ]}
         },
         {
            "id": "2",
            "action": "add",
            "state": "NOT_STARTED",
            "productOffering": {"id":
"http://serverlocation:port/catalogManagement/productOffering/43"},
            "product": {"productCharacteristics": [{
                  "name": "anotherCharacteristic",
```

```json
                "value": "itsValue"
            }]}
        }
    ]
}},
    "eventType": "orderCreationNotification"

}
```

OrderRemoveNotification

```json
{
    "event": {
        "productOrder": {
            "id": "http://serverlocation:port/orderManagement/productOrder/42",
            "externalID": "NiceNameForTheConsumer_42"
        },
        "reason" : "we do not keep order older than 1 year",
        "initiator" : "OM system",
        "date" : "2013-07-24 08:00:00ZGMT+1"
    },
    "eventType": "OrderRemoveNotification"
}
```

OrderValueChangeNotification

```json
{"event": {
  "productOrder": {
    "id": "http://serverlocation:port/orderManagement/productOrder/42",
    "externalID": "NiceNameForTheConsumer_42",
    "orderItems": [{
      "id": "1",
      "product": {"productCharacteristics": [{
        "name": "Colour",
        "value": "Black"
```

```
        }]}
    }],
    "reason": "customer requested change of colour",
    "initiator": "customer",
    "date": "2013-07-24 08:00:00ZGMT+1"
  },
  "eventType": "OrderValueChangeNotification"
}}
```

OrderStateChangeNotification

```
    {"event": {
      "productOrder": {
        "id": "http://serverlocation:port/orderManagement/productOrder/42",
        "externalID": "NiceNameForTheConsumer_42",
        "state": "CLOSE.COMPLETED",
      },
      "reason": "\u2026",
      "date": "2013-07-24 08:00:00ZGMT+1",
      "initiator": "OM system",
      "eventType": " orderStateChangeNotification "
    }}
```

tmf⌀rum

API OPERATION TEMPLATES

Operation on Entities	Uniform API Operation	Description
Query Entities	GET Resource	GET must be used to retrieve a representation of a resource.
Create Entity	POST Resource	POST must be used to create a new resource
Partial Update of an Entity	PATCH Resource	PATCH must be used to partially update a resource (Not covered in this 1st iteration)
Complete Update of an Entity	PUT Resource	PUT must be used to completely update a resource identified by its resource URI (Not covered in this 1st iteration)
Remove an Entity	DELETE Resource	DELETE must be used to remove a resource (Not covered in this 1st iteration)
Execute an Action on an Entity	POST on TASK Resource	POST must be used to execute Task Resources (maybe tomorrow?)
Other Request Methods	POST on TASK Resource	GET and POST must not be used to tunnel other

<Document Title>

| | | request methods. |
| | | (not useful |

Filtering and attribute selection rules are described in the TMF REST Design Guidelines.

Notifications are also described in a subsequent section.

GET /orderManagement/productOrder/{ID}

Note that collections can be retrieved via GET /API/<RESOURCE> with no {ID}

Description :

This operation is used to retrieve order information using the ID

Attribute selection is enabled on all first level attributes but not on inner classes.

Attribute selection may optionally be enabled on all attributes and inner classes.

productOrder identifiers are strings formatted as a URI.

Behavior :

- Return status codes
    - 200 OK - the request was successful
    - 404 Not found - the supplied ID does not match a known productOrder.

REQUEST
e.g. Getting a specific order identified by ID
GET /orderManagement/productOrder/42 Accept: application/json
RESPONSE
200 Content-Type: application/json [{"productOrder": {

tmforum

```json
"id": "http://serverlocation:port/orderManagement/productOrder/42",
"externalID": "NiceNameForTheConsumer_42",
"description": "A wonderful 42 order for brand new products",
"type": "residential",
"state": "CLOSED.COMPLETED",
"orderDate": "2013-07-23 08:16:39ZGMT+1",
"completionDate": "2013-07-24 01:00:00ZGMT+1",
"requestedCompletionDate": "2013-07-24 08:00:00ZGMT+1",
"relatedParties": [
    {
        "role": "customer",
        "reference": "MSISDN-0033689770600"
    },
    {
        "role": "partner",
        "reference": " http://serverlocation:port/partnerManagement/partner/42"
    }
],
"orderItems": [
    {
        "id": "1",
        "action": "add",
        "state": "COMPLETED",
        "productOffering": {"id":
"http://serverlocation:port/catalogManagement/productOffering/42"},
        "product": {"productCharacteristics": [
            {
                "name": "Colour",
                "value": "White"
            },
            {
                "name": "Memory",
                "value": "16"
            }
        ]}
    },
    {
        "id": "2",
        "action": "add",
        "state": "FAILED",
        "productOffering": {"id":
```

<Document Title>

```
"http://serverlocation:port/catalogManagement/productOffering/43"},
        "product": {"productCharacteristics": [{
          "name": "anotherCharacteristic",
          "value": "itsValue"
      }]}
    }
  ]
}}

]
```

GET /orderManagement/productOrder/{attributeSelection}?{query expression}

Note that collections can be retrieved via GET /API/<RESOURCE>

Description :

This operation is used to retrieve order information using filter criteria.

Filtering selection is enabled on all first level attributes but not on inner classes with the exception of relatedParties.

Filtering may optionally be enabled on all attributes and inner classes.

Attribute selection is enabled on all first level attributes but not on inner classes.

Attribute selection may optionally be enabled on all attributes and inner classes.

productOrder identifiers are strings formatted as a URI.

- • Return status codes
  - o 200 OK - the request was successful (includes situation in which no orders matched supplied criteria)
  - o 400 Bad Request - error

REQUEST

e.g. Getting all the orders for a given customer which were completed before a specified date

<Document Title>

tmforum

GET /orderManagement/productOrder? (relatedParties.role=customer& relatedParties.reference=
MSISDN-0033689770600) &completionDate.lt=2013-09-10 00:00:00ZGMT+1
Accept: application/json

RESPONSE

200
Content-Type: application/json

```
[{"productOrder": {
    "id": "http://serverlocation:port/orderManagement/productOrder/42",
    "externalID": "NiceNameForTheConsumer_42",
    "description": "A wonderful 42 order for brand new products",
    "type": "residential",
    "state": "CLOSED.COMPLETED",
    "orderDate": "2013-07-23 08:16:39ZGMT+1",
    "completionDate": "2013-07-24 01:00:00ZGMT+1",
    "requestedCompletionDate": "2013-07-24 08:00:00ZGMT+1",
    "relatedParties": [
        {
            "role": "customer",
            "reference": "MSISDN-0033689770600"
        },
        {
            "role": "partner",
            "reference": " http://serverlocation:port/partnerManagement/partner/42"
        }
    ],
    "orderItems": [
        {
            "id": "1",
            "action": "add",
            "state": "COMPLETED",
            "productOffering": {"id":
"http://serverlocation:port/catalogManagement/productOffering/42"},
            "product": {"productCharacteristics": [
                {
                    "name": "Colour",
                    "value": "White"
                },
                {
                    "name": "Memory",
                    "value": "16"
```

<Document Title>

```json
            }
        ]}
    },
    {
        "id": "2",
        "action": "add",
        "state": "FAILED",
        "productOffering": {"id":
"http://serverlocation:port/catalogManagement/productOffering/43"},
        "product": {"productCharacteristics": [{
            "name": "anotherCharacteristic",
            "value": "itsValue"
        }]}
    }
  ]
}}

]
```

PUT API/{RESOURCE}/{ID}

Not covered

REQUEST

PUT API/{RESOURCE}/{ID}
Content-type: application/json
 	
{
  JSON Resource Representation with every attribute
}

RESPONSE

201
Content-Type: application/json
 	
{
 JSON Resource Representation with every attribute
}

<Document Title>

Example see TMF REST Design Guidelines.

PATCH API/{RESOURCE}/{ID}

<mark>Not covered</mark>

This Uniform Contract operation is used to partially update the representation of a managed entity or a task.

Description :

- Provide an overall description of the Operation
- Describe the input representation of the <resource> instance.
- Describe if the resource represents a managed entity or a task.
- Describe the structure of the identifier.

Behavior :

- What status and exception codes are returned.
- Returns HTTP/1.1 status code 201 if the request was successful.
- Any other special return and/or exception codes.

REQUEST

PATCH API/{RESOURCE}/{ID}
Content-type: application/json
 	
{
  JSON Resource Representation with every attribute
}

RESPONSE
201 Content-Type: application/json { JSON Resource Representation with every attribute }

Example see TMF REST Design Guidelines.

POST ORDERMANAGEMENT/PRODUCTORDER

<Document Title>

Description :

This operation is used to create a new order.

The mandatory attributes for the creation are :

- Initial state is open.running if another state is provided at creation time then the order will be rejected.
- relatedParties.role and reference to identify at least the customer
- at least one orderItem with :
    o an action
    o a productOffering

Behavior :

- Return status codes
    o 201 Created - the request was successful
    o 400 Bad Request - error

REQUEST

```
POST orderManagement/productOrder
Content-type: application/json

{"productOrder": {
  "externalID": "NiceNameForTheConsumer_42",
  "description": "A wonderful 42 order for brand new products",
  "type": "residential",
  "requestedCompletionDate": "2013-07-24 08:00:00ZGMT+1",
  "relatedParties": [
    {
      "role": "customer",
      "reference": "MSISDN-0033689770600"
    },
    {
      "role": "partner",
      "reference": "http://serverlocation:port/partnerManagement/partner/42"
    }
  ],
  "orderItems": [
    {
      "action": "add",
      "productOffering": {"id":
"http://serverlocation:port/catalogManagement/productOffering/42"},
      "product": {"productCharacteristics": [
        {
          "name": "Colour",
```

<Document Title>

```json
                "value": "White"
            },
            {
                "name": "Memory",
                "value": "16"
            }
        ]}
    },
    {
        "action": "add",
        "productOffering": {"id":
"http://serverlocation:port/catalogManagement/productOffering/43"},
        "product": {"productCharacteristics": [{
            "name": "anotherCharacteristic",
            "value": "itsValue"
        }]}
    }
    ]
}}
```

RESPONSE

```json
201
Content-Type: application/json

{"productOrder": {
    "id": "http://serverlocation:port/orderManagement/productOrder/42",
    "externalID": "NiceNameForTheConsumer_42",
    "description": "A wonderful 42 order for brand new products",
    "type": "residential",
    "state": "OPEN.RUNNING",
    "orderDate": "2013-07-23 08:16:39ZGMT+1",
    "requestedCompletionDate": "2013-07-24 08:00:00ZGMT+1",
    "relatedParties": [
        {
            "role": "customer",
            "reference": "MSISDN-0033689770600"
        },
        {
            "role": "partner",
            "reference": " http://serverlocation:port/partnerManagement/partner/42"
        }
    ],
    "orderItems": [
        {
            "id": "1",
            "action": "add",
            "state": "NOT_STARTED",
```

<Document Title>

```
      "productOffering": {"id":
"http://serverlocation:port/catalogManagement/productOffering/42"},
      "product": {"productCharacteristics": [
        {
          "name": "Colour",
          "value": "White"
        },
        {

          "name": "Memory",
          "value": "16"
        }
      ]}
    },
    {
      "id": "2",
      "action": "add",
      "state": "",
      "productOffering": {"id":
"http://serverlocation:port/catalogManagement/productOffering/43"},
      "product": {"productCharacteristics": [{
        "name": "anotherCharacteristic",
        "value": "itsValue"
      }]}
    }
  ]
}}
```

Example see TMF REST Design Guidelines.

DELETE API/{RESOURCE}/{ID}

Not covered

This Uniform Contract operation is used to delete a managed entity or a task.

Description :

- Provide an overall description of the Operation
- Describe if the resource represents a managed entity or a task.
- Describe the structure of the identifier.

Behavior :

- What status and exception codes are returned.
- Returns HTTP/1.1 status code 200 if the request was successful.

<Document Title>

- Any other special return and/or exception codes.

REQUEST
DELETE API/{RESOURCE}/{ID}
RESPONSE
200

Example see TMF REST Design Guidelines.

API NOTIFICATION TEMPLATES

For every single of operation on the entities use the following templates and provide sample REST notification POST calls.

It is assumed that the Pub/Sub uses the Register and UnRegister mechanisms described in the REST Guidelines reproduced below.

REGISTER LISTENER POST /HUB

Description :

Sets the communication endpoint address the service instance must use to deliver information about its health state, execution state, failures and metrics. Subsequent POST calls will be rejected by the service if it does not support multiple listeners. In this case DELETE /api/hub/{id} must be called before an endpoint can be created again.

Behavior :

Returns HTTP/1.1 status code 204 if the request was successful.

Returns HTTP/1.1 status code 409 if request is not successful.

REQUEST
POST /api/hub Accept: application/json {"callback": "http://in.listener.com"}
RESPONSE
201 Content-Type: application/json Location: /api/hub/42 {"id":"42","callback":"http://in.listener.com","query":null}

<Document Title>

UNREGISTER LISTENER DELETE HUB/{ID}

Description :

Clears the communication endpoint address that was set by creating the Hub.

Behavior :

Returns HTTP/1.1 status code 204 if the request was successful.

Returns HTTP/1.1 status code 404 if the resource is not found.

REQUEST
DELETE /api/hub/{id} Accept: application/json
RESPONSE
204

PUBLISH {EVENTTYPE} POST /LISTENER

Description :

Provide the Event description

Behavior :

Returns HTTP/1.1 status code 201 if the service is able to set the configuration.

REQUEST

POST /client/listener
Accept: application/json
 	
{
 	
  "event": {

<Document Title>

```
   EVENT BODY
      },
   "eventType": "eventType"
}
```

RESPONSE

201
Content-Type: application/json

Example see TMF REST Design Guidelines.

<Document Title>

APPENDIX – PARKING LOT

The parking lot below lists changes made to the SID model for the purpose of simplifying the first cut of the interface, notes have been made where appropriate where entities were thought to be relevant in the future for API enabled updates.

PRODUCT ORDER

Parking Lot	
Can we add the ability to define custom events to be sent and received (specific to the application)?	Work order events may provide this functionality. To be determined.
Can we add the ability to define custom actions?	
Removed Quantity from BusinessInteractionItem	Might be added at a later stage
Status at the OrderItem level	Might be added at a later stage
Removed BusinessInteraction from BusinessInteractionItem	Might be added at a later stage
Removed BusinessInteractionRole from BusinessInteractionItem	Might be added at a later stage
Removed productSpecification from BusinessInteractionItem	Might be added at a later stage
Removed resourceSpecification from BusinessInteractionItem	Might be added at a later stage
Removed service from BusinessInteractionItem	Might be added at a later stage
Removed serviceSpecification from BusinessInteractionItem	Might be added at a later stage
BusinessInteractItem, BusinessInteractionItem1 removed from orderItem	Might be added at a later stage
BusinessInteractionLocation removed from orderItem	Might be added at a later stage
assignedPriority removed from Product Order	Might be added at a later stage
assignedResponsibilityDate removed from Product Order	Might be added at a later stage
BusinessInteractionLocation was removed	Might be added at a later stage
BusinessInteractionVersion was removed	Might be added at a later stage
BusinessInteractionSpec was removed	Might be added at a later stage

<Document #, Version #>

<Document Title>

Parking Lot	
BusinessInteractionType was removed	Might be added at a later stage
Place was removed from Product Order	Might be added at a later stage
BusinessInteraction, BusinessInteraction1-5 removed from Product Order (for bundling)	Might be added at a later stage
Removed dunningCaseRule	Might be added at a later stage

Content-range management in GET requests does not need to be managed for the hackathon.

<Document Title>

tmforum

APPENDIX – SID MAPPING

The SID mapping below lists changes made to the SID model for the purpose of improving readability of the interface.

PRODUCT ORDER

SID Mapping	
productOrder is defined by collapsing information from Product Order, BusinessInteraction, and BusinessInteractionItem entities	
Removed Product Order.BusinessInteractionRole	Replaced with productOrder.relatedParties
Product Order.interactionDate	Renamed to productOrder.orderDate
Product Order.interactionDateComplete	Renamed to productOrder.completionDate
Product Order.interactionStatus	Renamed to productOrder.status
Product Order.BusinessInteractionItem	Renamed to productOrder.orderItem
productSpecCharacteristics structure was renamed productCharacteristics	
productCharacteristics structure was simplified (only name and value are used)	
Product Order.customerOrderType	Renamed to productOrder.type
Product Order.purchaseOrderNumber	Renamed to productOrder.externalID
Renamed ID from BusinessInteractionItem	Renamed to productOrder.orderItem.name

<Document Title>

RELEASE HISTORY

Release Number	Date	Release led by:	Description
Release 1.0	tbd	Pierre Gauthier TM Forum pgauthier@tmforum.org	First Release of Draft Version of the Document.

<Document #, Version #>

<Document Title>

ACKNOWLEDGEMENTS

<Document #, Version #>