Release Notes for Cisco ServiceGrid 7.2

Revision History

<table>
<thead>
<tr>
<th>Version No.</th>
<th>Description</th>
<th>Created By / Modified By</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>Initial Draft version</td>
<td>Meenakshi Kamaraj</td>
</tr>
<tr>
<td>1.1</td>
<td>Final Draft. Updated with new images and incorporated the latest comments.</td>
<td>Gloria</td>
</tr>
</tbody>
</table>
Introduction

Cisco ServiceGrid is an integration platform in the cloud for IT service management. It provides a scalable, highly secure, and faster way to integrate with everyone in your service ecosystem, and also meets your business requirements. It creates operational efficiencies that save you time and money, while simplifying the formation of your ecosystem so that you can collaborate faster.

This document describes the key features associated with the ServiceGrid 7.2 Release.

This document contains the following sections:

- Release Dates, page 2
- System Requirements, page 3
- New and Enhanced Features in Cisco ServiceGrid Release 7.2, page 4
- Important Notes, page 25
- Limitations and Restrictions, page 25
- End-User License Agreement, page 25
- ServiceGrid Documentation, page 25
- Support Information, page 26

Release Dates

The Cisco ServiceGrid functions of the Release 2015 (Version 7.2) are available on July 19, 2015 to all customers using the Cisco ServiceGrid main platform (sdcall.solvedirect.com).

Release Notes 7.2 is in production on the Cisco ServiceGrid support platform from July 19, 2015. All customers running their own in-house infrastructure or using a Cisco partner infrastructure will receive the release on a later date. These updates will take place after the update of the Cisco ServiceGrid main platform. Contact your implementation partner for the date of your update.
System Requirements

Cisco ServiceGrid Online application (Portal, SD2) is a web-based application and hence is accessible using a browser. The B2B connection uses the ITSM connection capabilities of the customers.

The following browser versions were tested for Release 7.2 with respect to the browser classes:
- Firefox v37, v38
- Internet Explorer v10, v11
- Google Chrome v42, v43

Active SLA features should be used with the most recent versions of all browsers provided in Browser Class 1 in Table 1, and while using Internet Explorer, “compatibility mode” must be deactivated.

<table>
<thead>
<tr>
<th>Browser Class</th>
<th>Browser</th>
<th>Properties</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Mozilla Firefox (last two major versions)</td>
<td>Complete availability of product and application features (technician calendar, HTML-editor, and so on).</td>
</tr>
<tr>
<td></td>
<td>Google Chrome (last two major versions)</td>
<td>Graphical presentation (CSS layout).</td>
</tr>
<tr>
<td></td>
<td>Internet Explorer 11</td>
<td>No open browser-related known errors.</td>
</tr>
<tr>
<td>2</td>
<td>Internet Explorer 10</td>
<td>Limited availability of product and application features.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Limited graphical presentation (CSS Layout).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>There may be browser-related bugs/known errors.</td>
</tr>
<tr>
<td>3</td>
<td>Internet Explorer 9</td>
<td>Limited availability of product and application features.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Highly limited graphical presentation (CSS Layout).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Open browser-related bugs/known errors.</td>
</tr>
</tbody>
</table>

Note

Active SLA features should be used with the most recent versions of all browsers provided in Browser Class 1 in Table 1, and while using Internet Explorer, “compatibility mode” must be deactivated.
New and Enhanced Features in Cisco ServiceGrid
Release 7.2

The following features and enhancements are provided in Cisco ServiceGrid Release 7.2:

- **Rest APIs**
  - OAuth 2.0 Authentication
  - Call Services
- **Push (=putcall) a new call or a call update**
- **Pull (=getCalls) a new call or a call update**
- **Ticket Services**
  - Create a New Ticket
  - Update an Existing Ticket
  - Get a list of Tickets
  - Get the Details of One Ticket
- **Authentication Settings with Service Definition**
  - Setting Up Re-Authentication Request
- **B2B Connections**
  - ServiceGrid Standard JSON Converter
  - Custom JSON Converter
  - Platform Name Available for Error Messages
- **ServiceGrid Portal**
  - User Specific Size of Ticket Detail Window
  - List of Values for Extended Fields
- **Technology, Sub-technology, and Problem Codes**
  - Unique ID for TSP Codes
  - Pull all valid TSP codes from a REST API
Rest APIs

The Cisco ServiceGrid Release 7.2 has introduced the Rest API functionality to enable the customers to:

- access ticket data using the Rest API,
- send the ticket details to the Rest Service after the tickets are created or updated,
- retrieve changed tickets from the Rest Service.

Rest API service provides two new services for the exchange of tickets:

- Call Services
- Ticket Services

OAuth 2.0 Authentication

Starting Release 7.2, ServiceGrid supports the open standard or framework for authorization “OAuth 2.0”. It is a defacto-standard to provide client applications a “secure delegated access” to server resources on behalf of a resource owner. It specifies a process for resource owners to authorize third-party access to their server resources without sharing their credentials. OAuth essentially allows access tokens to be issued to third-party clients by an authorization server, with the approval of the resource owner. The client then uses the access token to access the protected resources hosted by the resource server.

Clients must use their ServiceGrid username and password for basic authentication to obtain the OAuth access token through the /ws/rest/oauth/token endpoint using “grant_type=client_credentials” as a query parameter.

Note

From release 7.2, the access token obtained from the ServiceGrid OAuth Server must be used to authenticate all REST services of ServiceGrid. The obtained token together with the token type must be sent as an “Authorization Header” parameter.

Call Services

Call services provides a new way of accessing the converter and queuing processes. It uses the following POST methods for an event-based asynchronous exchange of tickets (calls). A message in JSON format is accepted by push/Call and returned by pull/Call. In this release, the message is converted and further handled as an XML message. It can be sent to host:/port/ws/rest/v1/push/call or pulled from host:/port/ws/rest/v1/pull/call. This Rest API service resource works with basic authentication.

Push (=put call) a new call or a call update

This new API allows you to push, create and update messages to ServiceGrid using JSON as a message format.

API endpoint: host:/port/ws/rest/v1/push/call

In order to use the putCall operation, an appropriate inbound trigger must be configured. This inbound trigger needs to have a communication using:

- SDStandardJSON or CustomJsonToXmlConverter as converter and
- HTTPS POST as communication type (HTTPDirect)
Add a user with the permission group ‘SYS’ to the communication, which will be used for the authentication process. To create a call, a contract and a contract element must be transmitted.

Example for doing a test push of a message:

cat putcall.json | curl -X POST --user 'user:password' -H 'Content-Type: application/json' -d @- -i 'host:8080/ws/json/v1/push/call'

putcall.json:

```
{
    "Calls": {
        "Description": "Testticket with JSON",
        "SPCallID": "INC123456"
    },
    "Contracts": {
        "ShortName": "Contract"
    },
    "ContractElements": {
        "ShortName": "ContractElement"
    },
    "CallStates": {
        "ShortName": "INC01"
    }
}
```
Pull (=getCalls) a new call or a call update

This API allows you to pull open and update messages from ServiceGrid using JSON as a message format.

**API endpoint:** `host:port/ws/rest/v1/pull/call`

In order to use getCall, an appropriate outbound trigger must be configured. This outbound trigger must have a communication using:

- SDStandardJSON or CustomJsonToXmlConverter as converter and
- HTTPS POST as communication type (HTTPDirect)

**Note**

Add a user with the permission group ‘SYS’ to the communication, which will be used for the authentication process.

**Example for doing a test pull of a message:**

   curl -X POST --user 'user:password' -i 'host:8080/ws/json/v1/pull/call'
Ticket Services

Ticket Services is a new resource of Rest services using GET, POST and PATCH methods will allow you to push and pull create and update messages from ServiceGrid and will work with basic user authentication. All synchronous requests will go straight to the database, hence it is not necessary to configure any inbound or outbound triggers. This means that no transformation (for example, field mappings or business logic) of the JSON messages can be performed within ServiceGrid.

Create a New Ticket

A new ticket can be created with this Rest API with the method POST. All ticket attributes are available through this Rest API create operation.

**API endpoint:** `host:port/ws/rest/v1/tickets`

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Description</th>
<th>Parameter Type</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>contract</code>:</td>
<td><code>&quot;shortName&quot;: &quot;playground&quot;</code>, <code>&quot;contractElement&quot;: { &quot;shortName&quot;: &quot;playground&quot; },</code></td>
<td><code>description</code>: <code>&quot;description for case&quot;</code>, <code>&quot;customer/CustomerID&quot;: &quot;IDC_Customer_133929493&quot;</code>, <code>&quot;ticketState&quot;: { &quot;shortName&quot;: &quot;INCOL&quot; }</code></td>
<td>body</td>
<td>Model</td>
</tr>
</tbody>
</table>

**Note**
See online documentation at `hostname/ws (Rest API documentation)` for detailed information about the expected JSON format. All POST requests must be authenticated using an OAuth Bearer token in the Authorization HTTP header.
Update an Existing Ticket

ServiceGrid tickets can be updated through the Rest API using the HTTP PATCH operations such as add, remove, replace, copy and move and the JSON PATCH format.

**API endpoint:** `host:port/ws/rest/v1/tickets/{ID}`

This sample request would update the field “description” with the new value “overwrite description”.

```json
[{ "op": "replace", "path": "/description", "value" : "overwrite description" }]
```

**Implementation Notes**
Tickets can be updated via the REST API using the HTTP PATCH method and the JSON PATCH format as specified in RFC 6902.

**Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Description</th>
<th>Parameter Type</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>200271232</td>
<td></td>
<td>path</td>
<td>integer</td>
</tr>
<tr>
<td>body</td>
<td>[{ &quot;op&quot;: &quot;replace&quot;, &quot;path&quot;: &quot;/description&quot;, &quot;value&quot;: &quot;overwrite description with new value&quot; }]</td>
<td></td>
<td>body</td>
<td>Model</td>
</tr>
</tbody>
</table>

Parameter content-type: `application/json-patch+json`

**Request URL**


**Note**
A list of all attributes that can be changed through PATCH is retrieved through the online API documentation which is available at `hostname/ws` (Rest API documentation). All PATCH requests must be authenticated using an OAuth Bearer token in the Authorization HTTP header.
Get a list of Tickets

With the API method GET, a list of tickets including query parameters can be fetched from ServiceGrid.

**API endpoint:** `host:port/ws/rest/v1/tickets?offset=0&limit=1`

---

**Note**

v1 does not support “from-to” filters. Hence, filtering for timestamps and data fields might not be in the scope of this release. All GET requests need to be authenticated using an OAuth Bearer token in the Authorization HTTP header.
Get the Details of One Ticket

With this API method, the detailed data of a ticket can be fetched from ServiceGrid.

**API endpoint:** `host:port/ws/rest/v1/tickets{id}`
Authentication Settings with Service Definition

The new capabilities allow ServiceGrid to act as a REST client and support following HTTP verbs (POST, PUT, GET, PATCH and DELETE).

The primary use case for these “capabilites” is to fetch an authentication token from external REST service. If a token has been expired a new one can be easily automatically pulled.

It consists of the following extensions to the existing data model and user interface:

- A new field ‘AuthURL’ in the service definition allows the implementation specialist to specify the location of the authentication server. The ServiceGrid authentication URL would be, for example:
  
  https://at1.solvedirect.com/ws/rest/oauth/token

- A new field ‘AuthRegex’ allows you to specify how to extract the authentication data from a response received from the authentication server.

- A new macro of the form ${auth.response.regex.n} [where n stands for the capture groups in the regex] configured as the ‘AuthTokenExtractorRegex’ allows the implementation specialist to insert the authentication data dynamically into the final HTTP request (see example below).
• The new response filter action ‘Auth’ can be used to instruct ServiceGrid to retry authentication on certain conditions using the existing response filter capabilities to specify them.

**Change ResponseFilter master data:**

![Save] ![Cancel]

**master data**

<table>
<thead>
<tr>
<th>ID</th>
<th>200271019</th>
</tr>
</thead>
<tbody>
<tr>
<td>SeqNo</td>
<td>1</td>
</tr>
<tr>
<td>Match</td>
<td>All</td>
</tr>
<tr>
<td>HttpStatusCodeRange</td>
<td>401,403</td>
</tr>
<tr>
<td>MessageBodyRegex</td>
<td></td>
</tr>
<tr>
<td>Action</td>
<td>Auth</td>
</tr>
</tbody>
</table>

**Setting Up Re-Authentication Request**

To setup an OAuth2-based re-authenticated integration process, perform the following steps:

**Step 1**
Set up a new OB trigger (When update by) including a communication (Converter: SDStandardXSL).

**Step 2**
Add content template similar to the following example (see the comments in the example for details):

```xml
<SDHttpMessage>
  <!-- New authorization message element-->
  <AuthorizationMessage>
    <Headers>
      <!-- Will use basic authentication with username and password from the service definition unless specified otherwise -->
      <Parameter name="content-type">application/x-www-form-urlencoded</Parameter>
    </Headers>
    <Charset>UTF-8</Charset>
    <HttpMethod>POST</HttpMethod>
    <!-- new url suffix tag allows specification of query params -->
    <!-- as used typically in OAuth2 -->
    <UrlSuffix>grant_type=client_credentials</UrlSuffix>
    <Message/>
    </AuthorizationMessage>
    <Headers>
      <Parameter name="Content-Type">application/json</Parameter>
      <!-- results of applying the extractor regex are used here to authenticate the actual payload message -->
      <Parameter name="Authorization">Bearer ${auth.response.regex.1}</Parameter>
    </Headers>
    <Charset>UTF-8</Charset>
    <!-- Supported methods are now POST, GET, PUT and DELETE (with support for non-standard HTTP verbs coming soon)-->
    <HttpMethod>GET</HttpMethod>
    <PlainTextMessage>
```
Step 3  Add a Service Definition:

- Define Rest Service URL (Example: https://at1.solvedirect.com/ws/json/v1/push/call)
- Define the AuthUrl (Example: the ServiceGrid AuthUrl would be https://at1.solvedirect.com/ws/rest/oauth/token)
- Define UserName/Password combination to get an access token from the external OAuth server
- Define AuthRegex to extract the authentication data from a response received from the authentication server as the auth token. This Regex example:""access_token""s*:\s*""[^"\"]+"".* would work for the ServiceGrid authentication server and expose the access token in the first capture group (see the macro application ${auth.response.regex.1} above in the template.

Step 4  Add a response filter to the service definition to handle authentication failure and expiring authentication tokens. A typical filter would match on a HTTP response code 401 and would configure the new “Auth-Action”, which triggers a re-authentication attempt with the authentication server and resends the message with the newly obtained authentication data.

Note  Starting release 7.2, an administrator can override the target URL of the service definition with that of the SDHttpMessage. To accomplish this, the target URL can be defined within the SDHttpMessage with the new XML tag: <UrlSuffix>NewURL</UrlSuffix>.
# B2B Connections

## ServiceGrid Standard JSON Converter

A standard JSON format, similar to “SDStandardXML 2.0”, is introduced in the Cisco ServiceGrid release 7.2 to send ticket data in JSON format. In order to use this format within the communication, the standard “SDStandard JSON” converter must be configured.

### Example of a sample JSON message:

```json
{
  "Calls": {
    "Description": "Test ticket with Call-Service API - description",
    "SPCallID": "INC 12345678"
  },
  "Contracts": {
    "ShortName": "Contract"
  },
  "Contract Elements": {
    "ShortName": "Contract Element"
  },
  "Call States": {
    "ShortName": "01"
  }
}
```

### Note

Ensure that the existing system user is not used in any other communications which use an XML converter as this would cause XML parsing errors when sending JSON.
Custom JSON Converter

Starting release 7.2, a new inbound converter “CustomJsonToXmlConverter” is available for processing the messages sent in a custom JSON format. This converter translates the JSON messages into an XML message, which can then be used for further XSL translations.

Note

Message with an invalid JSON format will be rejected.

To test this new converter, messages can be sent to HTTPDirectServlet using username/password and the input message is sent as a body. The syntax is as shown below:

/ws/distinct/HTTPDirectServlet?SDUser=username&SDPassword=password

Each JSON message is converted into xml format using following rules:

1. The root element is always called “root” (root/root).
2. Fields are transformed into nested elements.
3. JSON field names which are not valid xml tag names are translated into names in which each invalid character is replaced with two underscores as prefix, then ordinal value of that character and one underscore as suffix.
4. Empty JSON names are translated into tag elements with three underscores as name.
5. JSON array is translated into nested elements with tag name ___ (three underscores).
6. Each xml element (including root elements) has type attribute whose value is “original type from JSON”, or there is NIL attribute with value “true” in case the original JSON field contains null values.
JSON examples with more special cases:

<table>
<thead>
<tr>
<th>JSON Format</th>
<th>XML Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>`{</td>
<td>&lt;root type=&quot;object&quot;&gt;</td>
</tr>
<tr>
<td>&quot;tickets&quot;: [</td>
<td>&lt;tickets type=&quot;array&quot;&gt;</td>
</tr>
<tr>
<td>{</td>
<td>&lt;___ type=&quot;object&quot;&gt;</td>
</tr>
<tr>
<td>&quot;id&quot;: 1234,</td>
<td>&lt;id type=&quot;number&quot;&gt;1234&lt;/id&gt;</td>
</tr>
<tr>
<td>&quot;external_id&quot;: &quot;9876ABC&quot;,</td>
<td>&lt;external_id type=&quot;text&quot;&gt;9876ABC&lt;/external_id&gt;</td>
</tr>
<tr>
<td>&quot;partner_ticket_id&quot;: &quot;P012&quot;,</td>
<td>&lt;partner_ticket_id type=&quot;text&quot;&gt;P012&lt;/partner_ticket_id&gt;</td>
</tr>
<tr>
<td>&quot;status&quot;: &quot;new&quot;,</td>
<td>&lt;status type=&quot;text&quot;&gt;new&lt;/status&gt;</td>
</tr>
<tr>
<td>&quot;title&quot;: &quot;Test Ticket&quot;,</td>
<td>&lt;title type=&quot;text&quot;&gt;Test Ticket&lt;/title&gt;</td>
</tr>
<tr>
<td>&quot;description&quot;: &quot;Test ticket with comment and attachment&quot;,</td>
<td>&lt;description type=&quot;text&quot;&gt;Test ticket with comment and attachment&lt;/description&gt;</td>
</tr>
<tr>
<td>&quot;severity&quot;: &quot;04&quot;,</td>
<td>&lt;severity type=&quot;text&quot;&gt;04&lt;/severity&gt;</td>
</tr>
<tr>
<td>&quot;impact&quot;: &quot;Low&quot;,</td>
<td>&lt;impact type=&quot;text&quot;&gt;Low&lt;/impact&gt;</td>
</tr>
<tr>
<td>&quot;services_impacted&quot;: [],</td>
<td>&lt;services_impacted type=&quot;array&quot;&gt;</td>
</tr>
<tr>
<td>&quot;assignee&quot;: &quot;Name&quot;,</td>
<td>&lt;assignee type=&quot;text&quot;&gt;Name&lt;/assignee&gt;</td>
</tr>
<tr>
<td>&quot;created&quot;: &quot;2015-06-01 08:33:45&quot;,</td>
<td>&lt;created type=&quot;text&quot;&gt;2015-06-01 08:33:45&lt;/created&gt;</td>
</tr>
<tr>
<td>&quot;created_by&quot;: &quot;Customer&quot;,</td>
<td>&lt;created_by type=&quot;text&quot;&gt;Customer&lt;/created_by&gt;</td>
</tr>
<tr>
<td>&quot;comments&quot;: [</td>
<td>&lt;comments type=&quot;array&quot;&gt;</td>
</tr>
<tr>
<td>{</td>
<td>&lt;___ type=&quot;object&quot;&gt;</td>
</tr>
<tr>
<td>&quot;author&quot;: &quot;Author&quot;,</td>
<td>&lt;author type=&quot;text&quot;&gt;Author&lt;/author&gt;</td>
</tr>
<tr>
<td>&quot;body&quot;: &quot;This is a comment&quot;,</td>
<td>&lt;body type=&quot;text&quot;&gt;This is a comment&lt;/body&gt;</td>
</tr>
<tr>
<td>&quot;created&quot;: &quot;2015-06-01 11:13:08&quot;</td>
<td>&lt;/___&gt;</td>
</tr>
<tr>
<td>}</td>
<td>&lt;/comments&gt;</td>
</tr>
<tr>
<td>}</td>
<td>&lt;/attachments&gt;</td>
</tr>
<tr>
<td>}</td>
<td>&lt;/___&gt;</td>
</tr>
<tr>
<td>`};</td>
<td>}</td>
</tr>
<tr>
<td>}</td>
<td>}</td>
</tr>
<tr>
<td>JSON Format</td>
<td>XML Format</td>
</tr>
<tr>
<td>----------------------------------------------------------------------------</td>
<td>----------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| `{                                                                       | <xml version="1.0" encoding="UTF-8"?>
|   "int": 1,                                                           | <root type="object">
|   "float": 1.4,                                                      |   <int type="number">1</int>
|   "string": "text",                                                  |   <float type="number">1.4</float>
|   "boolean": true,                                                   |   <string type="text">text</string>
|   "null": null,                                                      |   <boolean type="boolean">true</boolean>
|   "emptyArray": [],                                                  |   <null nil="true" />
|   "emptyObject": {},                                                 |   <emptyArray type="array" />
|   "object": {                                                        |   <emptyObject type="object" />
|     "inner": "innerValue"                                           |     <object type="object">
|   },                                                                  |       <inner type="text">innerValue</inner>
|   "mixedArray":[1,2.4,"a",true,null,{},{"x":0}][],                   |     </object>
|   "0name": "name with digit as first character",                    |   <mixedArray type="array">
|   "": "empty name",                                                  |     <___ type="number">1</___>
|   ":": "colon name",                                                 |     <___ type="number">2.4</___>
|   "-@#$": "some invalid tagname characters"                           |     <___ type="text">a</___>
| }                                                                       |     <___ type="boolean">true</___>
|                                                                          |     <___ nil="true" />
|                                                                          |     <___ type="object" />
|                                                                          |     <___ type="object">
|                                                                          |       <x type="number">0</x>
|                                                                          |     </___>
|                                                                          |   </mixedArray>
|                                                                          |   <__30_name type="text">name with digit as first character</__30_name>
|                                                                          |     <___ type="number">1</___>
|                                                                          |     <___ type="text">empty name</___>
|                                                                          |     <___ type="text">colon name</___>
|                                                                          |     <___ type="number">0</___>
|                                                                          |     <___ type="number">2.4</___>
|                                                                          |     <___ type="text">a</___>
|                                                                          |     <___ type="boolean">true</___>
|                                                                          |     <___ nil="true" />
|                                                                          |     <___ type="object" />
|                                                                          |     <___ type="object">
|                                                                          |       <x type="number">0</x>
|                                                                          |     </___>
|                                                                          |   </mixedArray>
|                                                                          |   <__30_name type="text">name with digit as first character</__30_name>
|                                                                          |     <___ type="number">1</___>
|                                                                          |     <___ type="text">empty name</___>
|                                                                          |     <___ type="text">colon name</___>
|                                                                          |     <___ type="number">0</___>
|                                                                          |     <___ type="number">2.4</___>
|                                                                          |     <___ type="text">a</___>
|                                                                          |     <___ type="boolean">true</___>
|                                                                          |     <___ nil="true" />
|                                                                          |     <___ type="object" />
|                                                                          |     <___ type="object">
|                                                                          |       <x type="number">0</x>
|                                                                          |     </___>
|                                                                          |   </mixedArray>
|                                                                          |   <__30_name type="text">name with digit as first character</__30_name>
|                                                                          |     <___ type="number">1</___>
|                                                                          |     <___ type="text">empty name</___>
|                                                                          |     <___ type="text">colon name</___>
|                                                                          |     <___ type="number">0</___>
|                                                                          |     <___ type="number">2.4</___>
|                                                                          |     <___ type="text">a</___>
|                                                                          |     <___ type="boolean">true</___>
|                                                                          |     <___ nil="true" />
|                                                                          |     <___ type="object" />
|                                                                          |     <___ type="object">
|                                                                          |       <x type="number">0</x>
|                                                                          |     </___>
|                                                                          |   </mixedArray>
|                                                                          |   <__30_name type="text">name with digit as first character</__30_name>
|                                                                          |     <___ type="number">1</___>
|                                                                          |     <___ type="text">empty name</___>
|                                                                          |     <___ type="text">colon name</___>
|                                                                          |     <___ type="number">0</___>
|                                                                          |     <___ type="number">2.4</___>
|                                                                          |     <___ type="text">a</___>
|                                                                          |     <___ type="boolean">true</___>
|                                                                          |     <___ nil="true" />
|                                                                          |     <___ type="object" />
|                                                                          |     <___ type="object">
|                                                                          |       <x type="number">0</x>
|                                                                          |     </___>
|                                                                          |   </mixedArray>
|                                                                          |   <__30_name type="text">name with digit as first character</__30_name>
|                                                                          |     <___ type="number">1</___>
|                                                                          |     <___ type="text">empty name</___>
|                                                                          |     <___ type="text">colon name</___>
|                                                                          |     <___ type="number">0</___>
|                                                                          |     <___ type="number">2.4</___>
|                                                                          |     <___ type="text">a</___>
|                                                                          |     <___ type="boolean">true</___>
|                                                                          |     <___ nil="true" />
|                                                                          |     <___ type="object" />
|                                                                          |     <___ type="object">
|                                                                          |       <x type="number">0</x>
|                                                                          |     </___>
|                                                                          |   </mixedArray>
|                                                                          |   <__30_name type="text">name with digit as first character</__30_name>
|                                                                          |     <___ type="number">1</___>
|                                                                          |     <___ type="text">empty name</___>
|                                                                          |     <___ type="text">colon name</___>
|                                                                          |     <___ type="number">0</___>
|                                                                          |     <___ type="number">2.4</___>
|                                                                          |     <___ type="text">a</___>
|                                                                          |     <___ type="boolean">true</___>
|                                                                          |     <___ nil="true" />
|                                                                          |     <___ type="object" />
|                                                                          |     <___ type="object">
|                                                                          |       <x type="number">0</x>
|                                                                          |     </___>
|                                                                          |   </mixedArray>
|                                                                          |   <__30_name type="text">name with digit as first character</__30_name>
|                                                                          |     <___ type="number">1</___>
|                                                                          |     <___ type="text">empty name</___>
|                                                                          |     <___ type="text">colon name</___>
|                                                                          |     <___ type="number">0</___>
|                                                                          |     <___ type="number">2.4</___>
|                                                                          |     <___ type="text">a</___>
|                                                                          |     <___ type="boolean">true</___>
|                                                                          |     <___ nil="true" />
|                                                                          |     <___ type="object" />
|                                                                          |     <___ type="object">
|                                                                          |       <x type="number">0</x>
|                                                                          |     </___>
|                                                                          |   </mixedArray>
|                                                                          |   <__30_name type="text">name with digit as first character</__30_name>
|                                                                          |     <___ type="number">1</___>
|                                                                          |     <___ type="text">empty name</___>
|                                                                          |     <___ type="text">colon name</___>
|                                                                          |     <___ type="number">0</___>
|                                                                          |     <___ type="number">2.4</___>
|                                                                          |     <___ type="text">a</___>
|                                                                          |     <___ type="boolean">true</___>
|                                                                          |     <___ nil="true" />
|                                                                          |     <___ type="object" />
|                                                                          |     <___ type="object">
|                                                                          |       <x type="number">0</x>
|                                                                          |     </___>
|                                                                          |   </mixedArray>
|                                                                          |   <__30_name type="text">name with digit as first character</__30_name>
|                                                                          |     <___ type="number">1</___>
|                                                                          |     <___ type="text">empty name</___>
|                                                                          |     <___ type="text">colon name</___>
|                                                                          |     <___ type="number">0</___>
|                                                                          |     <___ type="number">2.4</___>
|                                                                          |     <___ type="text">a</___>
|                                                                          |     <___ type="boolean">true</___>
|                                                                          |     <___ nil="true" />
|                                                                          |     <___ type="object" />
|                                                                          |     <___ type="object">
|                                                                          |       <x type="number">0</x>
|                                                                          |     </___>
|                                                                          |   </mixedArray>
|                                                                          |   <__30_name type="text">name with digit as first character</__30_name>
|                                                                          |     <___ type="number">1</___>
|                                                                          |     <___ type="text">empty name</___>
|                                                                          |     <___ type="text">colon name</___>
|                                                                          |     <___ type="number">0</___>
|                                                                          |     <___ type="number">2.4</___>
|                                                                          |     <___ type="text">a</___>
|                                                                          |     <___ type="boolean">true</___>
|                                                                          |     <___ nil="true" />
|                                                                          |     <___ type="object" />
|                                                                          |     <___ type="object">
|                                                                          |       <x type="number">0</x>
|                                                                          |     </___>
|                                                                          |   </mixedArray>
|                                                                          |   <__30_name type="text">name with digit as first character</__30_name>
|                                                                          |     <___ type="number">1</___>
|                                                                          |     <___ type="text">empty name</___>
|                                                                          |     <___ type="text">colon name</___>
|                                                                          |     <___ type="number">0</___>
|                                                                          |     <___ type="number">2.4</___>
|                                                                          |     <___ type="text">a</___>
|                                                                          |     <___ type="boolean">true</___>
|                                                                          |     <___ nil="true" />
|                                                                          |     <___ type="object" />
|                                                                          |     <___ type="object">
|                                                                          |       <x type="number">0</x>
|                                                                          |     </___>
|                                                                          |   </mixedArray>
### Platform Name Available for Error Messages

When using XSLT template in the “WhenErrorOccurs” trigger, the element “/SD.call/Error/SystemName” provides the access to the platform name (sdcall, sjc1, eu2-dev, ...). This parameter can be used for setting up business logic based on this value in this template.

```
eu2-dev / d3adm

Error-Message:

-----------------------------------------------
SolveDirect.com
Customized Error message

-----------------------------------------------
An error occured in your call-message
TRANSACTION REJECTED!

ERROR: CallStates ShortName 'I01x' in Callsystem Incident unknown!

-----------------------------------------------
CustCallID:
SDCallID2: 522416891
SPCallID:
Contract: Steer-INC
ContractElement: INC-Hardware
ErrorState: REJ_ERROR
Platform: eu2-dev
```
ServiceGrid Portal

User Specific Size of Ticket Detail Window

A new parameter has been added for capturing the current page size of the Cisco ServiceGrid portal. If the user is modifying the size of the ticket detail window, the new size will be retained even after a new login.

List of Values for Extended Fields

With the new release 7.2, the limitation of the field length of extended fields ExtTableValues/[Field1-128] has been changed from 50 to 255 characters. This feature can be useful if some special fields (CI, Assignment Group, and so on) are available in the service portal through a drop-down menu and the character limit for these drop-down fields be set to 255.

Step 1
Define the table extension and the value lists.

```
<table>
<thead>
<tr>
<th>TableExtensionField master data select a function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display values</td>
</tr>
<tr>
<td>Display categories</td>
</tr>
<tr>
<td>Change TableExtensionField master data</td>
</tr>
<tr>
<td>Delete TableExtensionField</td>
</tr>
<tr>
<td>Back to list</td>
</tr>
</tbody>
</table>

longdrowdown master data

| Id: | 2002710025 |
| FieldIndex: | 1 |
| ShortName: | longdrowdown |
| Label: | longdrowdown |
| DataType: | String |
| ValueFormat: | 1 |
| ValueList: | this is a very long value for this tableextension field with a size of 255 characters |
| AddEmptyEntry: | |
| InputTypes: | |
```

Step 2
Assign this table extension to a call detail.
TableExtensionField master data select a function

- Display values
- Display categories
- Change TableExtensionField master data
- Delete TableExtensionField
- Back to list

<table>
<thead>
<tr>
<th>longdropdown master data</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
</tr>
<tr>
<td>FieldIndex</td>
</tr>
<tr>
<td>ShortName</td>
</tr>
<tr>
<td>Label</td>
</tr>
<tr>
<td>DataType</td>
</tr>
<tr>
<td>ValueFormat</td>
</tr>
<tr>
<td>ValueList</td>
</tr>
<tr>
<td>AddEmptyEntry</td>
</tr>
<tr>
<td>InputType</td>
</tr>
</tbody>
</table>
**Step 3**  Open CallDetail within service portal.

<table>
<thead>
<tr>
<th>PROVIDER</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Provider</strong></td>
<td>Playground_REST INC</td>
</tr>
<tr>
<td>Helpdesk</td>
<td>playground_rest</td>
</tr>
<tr>
<td><strong>Editor</strong></td>
<td>playground_rest</td>
</tr>
<tr>
<td>Remarks</td>
<td>New Incident - 2nd ticket</td>
</tr>
<tr>
<td>Diagnosis</td>
<td>New Incident - 2nd ticket</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LOCATION</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Country</strong></td>
<td>AUSTRIA</td>
</tr>
<tr>
<td>Zip</td>
<td>TBD</td>
</tr>
<tr>
<td>City</td>
<td>Innsbruck</td>
</tr>
<tr>
<td>Street</td>
<td>TBD</td>
</tr>
</tbody>
</table>

| DEVICE |  |
| ESCALATION |  |
| PLAN |  |

<table>
<thead>
<tr>
<th>SOLUTION</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>longdropdown</td>
<td>this_is_a_very_long_value_for_this_table_extensions_field_with_a_size_of_255_characters</td>
</tr>
<tr>
<td>longdropdown2</td>
<td>this_is_a_very_long_value_for_this_table_extensions_field_with_a_size_of_255_characters</td>
</tr>
</tbody>
</table>

Save
Technology, Sub-technology, and Problem Codes

Unique ID for TSP Codes

The T/ST/P codes can be pulled with the new release 7.2 through REST-based service. Therefore, three additional fields have been added to this service:

- **Existing Fields:**
  - TechName: former field TechnologyCode
  - SubTechName: former field SubTechnologyCode
  - ProblemCodeDescription: former field problem code

- **Additional Fields:**
  - TechId: new field, the unique ID of a technology code
  - SubTechId: new field, the unique ID of a sub technology code
  - ProblemCodeName: new field, the unique key of one problem code

**Note**

By installing this change, all existing codes are deleted and new codes are created in the first pull. This means that all codes will be marked as **Inserted** and will be sent to each partner.

Pull all valid TSP codes from a REST API

There are already multiple ways of how SG partners can access the list of Cisco TSP codes which are stored in SG (mail, ftp, pull, and so on). With the release 7.2, it is now possible to pull these codes through a REST API with the method GET.

**API endpoint:** `host:port/ws/rest/v1/tsp-codes`
New and Enhanced Features in Cisco ServiceGrid Release 7.2

Use the query parameters ‘offset’ and ‘limit’ for defining the range of TSP codes user wants to access/download. The API will always deliver back all valid codes.

**TspCodes**

**GET** /v1/tsp-codes

**Implementation Notes**
List all tspCodes. Currently no filter criteria are supported. For a complete list of all possible fields have a look at the release notes.

**Response Class**
**Model** | **Model Schema**

```json
{
  "tspCodes": [
  {
    "changeLog": "",
    "editTimeUtc": "",
    "id": 0,
    "problemCodeDescription": "",
    "problemCodeName": "",
    "subTechId": 0,
    "subTechName": ""
  }
  ]
}
```

**Response Content Type** `application/json`

**Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Description</th>
<th>Parameter Type</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>offset</td>
<td>0</td>
<td></td>
<td>query</td>
<td>integer</td>
</tr>
<tr>
<td>limit</td>
<td>1</td>
<td></td>
<td>query</td>
<td>integer</td>
</tr>
</tbody>
</table>

**Request URL**

`https://eu2-qa.servicegrid.cisco-ccs.com:443/we/rest/v1/tsp-codes?offset=0&limit=1`

---

**Note**

Users who are able to access the TSP Code list are defined using the CiscoPartners table.
Important Notes

- Cisco ServiceGrid will be deployed in the setup of all customers running their own in-house infrastructure or using a Cisco Partner infrastructure after ServiceGrid is deployed in the main platform.
- To know about the release date of Cisco ServiceGrid 7.2 deployment in their setup, the customers need to contact their implementation partner.

Limitations and Restrictions

The following provides the limitations and restrictions in Cisco ServiceGrid Release 7.2:

- The requirements mentioned in Browser Class 1 in “System Requirements” section on page 3 provide the minimum system requirements for Cisco ServiceGrid.

End-User License Agreement

All new functions and modules are installed on the corresponding platforms. New functions and modules which are part of the general update are available to all customers of that platform. Some of the new functions and modules must have their license before they are used in customized systems.

ServiceGrid Documentation

<table>
<thead>
<tr>
<th>Table 2</th>
<th>ServiceGrid Documentation</th>
</tr>
</thead>
</table>
Support Information

Table 3 Support Information

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco Support</td>
<td>email: <a href="mailto:tac@cisco.com">tac@cisco.com</a></td>
</tr>
<tr>
<td></td>
<td>Web: <a href="http://www.cisco.com/support">www.cisco.com/support</a></td>
</tr>
<tr>
<td>Customer/Partner Maintenance Announcements</td>
<td><a href="mailto:servicegrid-support@cisco.com">servicegrid-support@cisco.com</a></td>
</tr>
</tbody>
</table>

Related Articles

**APIs in Cisco ServiceGrid**

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: [www.cisco.com/go/trademarks](http://www.cisco.com/go/trademarks). Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1110R)

Release Notes for ServiceGrid 7.2 Release

© 2015 Cisco Systems, Inc. All rights reserved.