Confirm®

IoT Connector Schema

v19.00a.AM
Information in this document is subject to change without notice and does not represent a commitment on the part of the vendor or its representatives. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, without the written permission of Pitney Bowes Inc.

© 2018 Pitney Bowes. All rights reserved. MapInfo, the MapInfo logo and Confirm are trademarks of Pitney Bowes and/or its affiliates.

Corporate contact details for all offices can be found here:

Technical Support contact details can be found here:
https://www.pitneybowes.com/us/contact-dcs.html

Products named herein may be trademarks of their respective manufacturers and are hereby recognized. Trademarked names are used editorially, to the benefit of the trademark owner, with no intent to infringe on the trademark.

Open Source Attribution Notice

The Confirm suite of products contain the following open source software:

- Feature Data Objects v 3.5.0, which is licensed under GNU Lesser General Public License, Version 2.1, February 1999 with the unRAR restriction. The license can be downloaded from: http://fdo.osgeo.org/licenceAndGovernance.html. The source code for this software is available from http://fdo.osgeo.org/content/fdo-350-downloads

- MrSID software (specifically the mrsid32.dll) is used under license and is Copyright © 1995-2002, LizardTech, Inc., 1008 Western Ave., Suite 200, Seattle, WA 98104. All rights reserved. MrSID is protected by U.S. Patent No. 5,710,835. Foreign patents are pending. Unauthorized use or duplication prohibited.

Patented technology in the Software was developed in part through a project at the Los Alamos National Laboratory, funded by the U.S. Government and managed by the University of California. The U.S. Government has reserved rights in the technology, including a non-exclusive, nontransferable, irrevocable, paid-up license to practice or have practiced throughout the world, for or on behalf of the United States, inventions covered by the patent, and has other rights under 35 U.S.C. § 200-212 and applicable implementing regulations.

For further information, contact Lizardtech.

- NodaTime, version number 1.3.10, which is licensed under the Apache license, version number 2.0. The license can be downloaded from http://www.apache.org/licenses/LICENSE-2.0. The source code for this software is available from http://nodatime.org/.

- Chromium Embedded Framework, version 3, which is licensed under the New BSD License. The license can be downloaded from http://opensource.org/licenses/BSD-3-Clause. The source code for this software is available from https://code.google.com/p/chromiumembedded/downloads/list.

- Xilium.CefGlue, version 3, which is licensed under the MIT License (with portions licensed under the New BSD License). The licenses can be downloaded from https://github.com/mbostock/d3/blob/master/LICENSE. The source code for this software is available from https://xilium.bitbucket.org/cefglue/.

- D3 Data Driven Documentation, version 3.4.1, which is licensed under the New BSD License. The license can be downloaded from https://github.com/mbostock/d3/blob/master/LICENSE. The source code for this software is available from https://d3js.org/.

- OpenLayers, version 2.12, which is licensed under the Modified BSD License. The license can be downloaded from http://svn.openlayers.org/trunk/openlayers/license.txt. The source code for this software is available from http://trac.osgeo.org/openlayers/browser.

- OpenLayers, version 3, which is licensed under the BSD 2-Clause Licence. The license which can be downloaded from https://github.com/openlayers/ol3/blob/master/LICENSE.md. The source code for this software is available from https://github.com/openlayers/ol3.
• Proj4js, version 1+, which is licensed under the Apache License, Version 2, January 2004. The license can be downloaded from http://www.apache.org/licenses/LICENSE-2.0.html. The source code for this software is available from http://trac.osgeo.org/proj4js/.

• requireJS, version 2.1.2, which is licensed under the MIT License or the New BSD License. The license can be downloaded from https://github.com/jrburke/requirejs/blob/master/LICENSE. The source code for this software is available from http://requirejs.org/.

• Apache Cordova, version 4.2.0, which is licensed under the Apache License, Version 2, January 2004. The license can be downloaded from http://www.apache.org/licenses/LICENSE-2.0.html. The source code for this software is available from http://phonegap.com/download/.

December 10, 2018
# Table of Contents

## Specifications

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>IoT Connector XML Schema</td>
<td>6</td>
</tr>
<tr>
<td>Methods</td>
<td>6</td>
</tr>
</tbody>
</table>
Specifications

The following sections outline all the Specifications that exist within the Confirm functionality.

In this section

IoT Connector XML Schema 6
IoT Connector XML Schema

Introduction
This document describes the XML schema used to allow an external system or website to update Feature Conditions and check for existing Defects and create new ones, using the Confirm Connector. The data transfer will be initiated by the external system or website. The Connector will always send data as a reply to the request.

Operation Overview
The Request sent to the Connector contains an “Authentication” section detailing the database connection and Confirm User information, and an “Operation” section detailing the operations to be carried out on that database.

The rest of this document describes briefly each of the operations that might be included in a Request, and the corresponding Response for each.

Where data is provided in a lookup field it must correspond to a valid record in the Confirm database, otherwise a Fault will be returned.

Where subsequent sections refer to a “User” this is the Connector User, i.e. the Login Name the Connector uses in connecting to the Confirm database. This determines which data will be returned where Data Key Security is in place.

XML Schema Definitions
The Request XML is defined in the following schema documents:
- IoT.xsd

The Response XML is defined with the methods in the following section.

Methods
This section describes the Requests to be sent from the external system or website to Confirm, and what the Responses will contain.

FeatureUpdate > FeatureCondition
This operation is used to update Feature Conditions in Confirm. This method does not create history and is designed to be called frequently without storing additional data in Confirm.

A single request can update multiple Features each having multiple Observation Types with a Value specified.

Any validation errors (e.g. the Value not matching the Grade min or max conditions) are returned as a Warnings collection in the response.

Schema validation errors are returned as Connector Fault Responses.
GetDefectsForFeature
Returns the outstanding Defects for a Feature. This is intended to be used by external systems to prevent the creation of duplicate Defects.

Defects matching the following criteria are returned:
• Defects not linked to a Job that have Action Required set on the Defect Status
• Defects linked to a Job where the Actual Completion Date is not set and the Job Status is Outstanding

Schema validation errors are returned as Connector Fault Responses.

GetDefectsForFeature Response
When Outstanding Defects are found for the requested Feature the following information is returned.

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>DefectNumber</td>
<td>Decimal</td>
<td></td>
</tr>
<tr>
<td>DefectDate</td>
<td>DateTime</td>
<td></td>
</tr>
<tr>
<td>DefectTypeCode</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>DefectTypeName</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>DefectEastings</td>
<td>Decimal</td>
<td>Coordinates are for the defect itself.</td>
</tr>
<tr>
<td>DefectNorthing</td>
<td>Decimal</td>
<td>Coordinates are for the defect itself.</td>
</tr>
</tbody>
</table>

FeatureUpdate > FeatureDefect
This operation is used to create Defects in Confirm.

A single request can update multiple Features each having multiple Defects.

Any validation errors are returned as a Warnings collection in the response.

Schema validation errors are returned as Connector Fault Responses.