

# Integration Server Package for Zementis

User Guide

Version 10.3  
October 2018

---

This document applies to *Integration Server Package for Zementis 10.3* and to all subsequent releases.

Specifications contained herein are subject to change and these changes will be reported in subsequent release notes or new editions.

Copyright © 2018 Software AG, Darmstadt, Germany and/or Software AG USA Inc., Reston, VA, USA, and/or its subsidiaries and/or its affiliates and/or their licensors.

The name Software AG and all Software AG product names are either trademarks or registered trademarks of Software AG and/or Software AG USA Inc. and/or its subsidiaries and/or its affiliates and/or their licensors. Other company and product names mentioned herein may be trademarks of their respective owners.

Detailed information on trademarks and patents owned by Software AG and/or its subsidiaries is located at <http://softwareag.com/licenses>.

Use of this software is subject to adherence to Software AG's licensing conditions and terms. These terms are part of the product documentation, located at <http://softwareag.com/licenses/> and/or in the root installation directory of the licensed product(s).

This software may include portions of third-party products. For third-party copyright notices, license terms, additional rights or restrictions, please refer to "License Texts, Copyright Notices and Disclaimers of Third Party Products". For certain specific third-party license restrictions, please refer to section E of the Legal Notices available under "License Terms and Conditions for Use of Software AG Products / Copyright and Trademark Notices of Software AG Products". These documents are part of the product documentation, located at <http://softwareag.com/licenses> and/or in the root installation directory of the licensed product(s).

DOCUMENT ID: [ZIS-UG-20181015]

## Contents

<b>1</b>	<b>Introduction</b> .....	<b>5</b>
<b>2</b>	<b>Setup</b> .....	<b>6</b>
2.1	Requirements .....	6
2.2	Installation .....	6
2.3	Configuration.....	6
<b>3</b>	<b>Services</b> .....	<b>7</b>
3.1	Namespaces .....	7
3.2	Model.....	9
3.2.1	<i>pub.zementis.model:Field</i> .....	9
3.2.2	<i>pub.zementis.model:ModelInfo</i> .....	9
3.2.3	<i>pub.zementis.model:listModels</i> .....	9
3.2.4	<i>pub.zementis.model:modelExists</i> .....	9
3.2.5	<i>pub.zementis.model:getModelInfo</i> .....	10
3.2.6	<i>pub.zementis.model:uploadModel</i> .....	10
3.2.7	<i>pub.zementis.model:getModelSource</i> .....	10
3.2.8	<i>pub.zementis.model:deleteModel</i> .....	11
3.2.9	<i>pub.zementis.model:deleteAllModels</i> .....	11
3.2.10	<i>wm.zementis.model:activateModel</i> .....	11
3.2.11	<i>wm.zementis.model:deactivateModel</i> .....	11
3.2.12	<i>wm.zementis.model:getSerializedModel</i> .....	11
3.3	Apply .....	12
3.3.1	<i>pub.zementis.apply:Record</i> .....	12
3.3.2	<i>pub.zementis.apply:recordApply</i> .....	12
3.3.3	<i>pub.zementis.model:batchApply</i> .....	12
3.4	Resource .....	13
3.4.1	<i>pub.zementis.resource:ResourceInfo</i> .....	13
3.4.2	<i>pub.zementis.resource:listResources</i> .....	13
3.4.3	<i>pub.zementis.resource:resourceFileExists</i> .....	13
3.4.4	<i>pub.zementis.resource:getResourceInfo</i> .....	13
3.4.5	<i>pub.zementis.resource:uploadResource</i> .....	14
3.4.6	<i>pub.zementis.resource:getResourceFile</i> .....	14
3.4.7	<i>pub.zementis.resource:deleteResource</i> .....	14
3.4.8	<i>pub.zementis.resource:deleteAllResources</i> .....	15
3.5	Configuration.....	15
3.5.1	<i>wm.zementis.config:getConfig</i> .....	15

3.5.2	<code>wm.zementis.config:setConfig</code> .....	15
3.5.3	<code>wm.zementis.config:testConnection</code> .....	15
3.5.4	<code>wm.zementis.config:startup</code> .....	16
3.5.5	<code>wm.zementis.config:shutdown</code> .....	16
3.6	Samples .....	16
3.6.1	<code>sample.zementis:singleRecordApply</code> .....	16
3.6.2	<code>sample.zementis:batchApply</code> .....	16
3.6.3	<code>sample.zementis:batchApplyWithResources</code> .....	16
3.6.4	<code>sample.zementis:updateModel</code> .....	17
3.6.5	<code>sample.zementis:updateResource</code> .....	17
3.6.6	<code>sample.zementis:pmmlToDocument</code> .....	17

## 1 Introduction

**Zementis Predictive Analytics (Zementis)** product suite enables agile deployment and integration of machine learning and predictive analytics models. **Zementis** is based on open standards **Predictive Model Markup Language (PMML)** to facilitate interoperability of predictive models across applications and vendors. **Zementis** establishes a common, standards-based framework to deliver intelligent solutions across all industries and applications. With **Zementis**, organizations can harness the power of their data to rapidly deliver insights and support informed business decisions based on predictive analytics. **Zementis** ensures model integrity, optimizes performance, enables scaling as necessary and allows scoring data in real time and batch mode.

**Integration Server Package for Zementis (WmPredictiveAnalytics)** package acts as a client to **Zementis Server** and enables easy integration with wide array of protocols, adapters, flow services, or any other services supported by **webMethods Integration Server (Integration Server)**.

To learn more about **Zementis Predictive Analytics**, please see included documentation, samples, and other resources on **Zementis Server** under `/adapaconsole/html/help.html#references` context path.

## 2 Setup

### 2.1 Requirements

**Zementis Server** instance must be running and be accessible from **Integration Server** instance environment via **HTTP** or **HTTPS** protocol for user with administrative access privileges.

### 2.2 Installation

The **WmPredictiveAnalytics** package can be installed using **Software AG Installer**. The package does not require any license.

### 2.3 Configuration

Before executing services available with **WmPredictiveAnalytics** package, access information to **Zementis Server** needs to be configured on **Integration Server**.

This can be done via **Integration Server Administrator Utility** web application. Navigate to Packages->Management web page and click home icon next in **WmPredictiveAnalytics** package listing. You can also append package name to base **URL** in browser's address window, for example <http://localhost:5555/WmPredictiveAnalytics>.

Once on package configuration page, enter host **URL** for **Zementis Server**, for example <http://localhost:9083>, or <https://mydomain.com>, then enter user name for user with administrative access to **Zementis Server**, and password. Click on **Test Connection** to test the username and password, on receiving a successful connection message click on **Save Configuration** button and you should see confirmation message displayed.

Zementis Client Configuration	
Host URL	<input type="text" value="http://localhost:9083"/>
Username	<input type="text" value="Administrator"/>
Password	<input type="password"/>
<input type="button" value="Test Connection"/> <input type="button" value="Save Configuration"/> <input type="button" value="Clear"/>	

## 3 Services

Functionally, package services can be categorized as [model management](#), [data processing](#), [model resources management](#), [package configuration](#), and [flow samples](#).

### 3.1 Namespaces

Services in **WmPredictiveAnalytics** are logically divided into three main namespaces: public, internal, and samples.

Public namespace, designated by 'pub.zementis' prefix, lists services for most common operations on PMML models, resources, and applying data to uploaded model.

Internal namespace, with 'wm.zementis' prefix, contains package configuration services, and model services that are meant for internal use. Samples depict flow services which use **WmPredictiveAnalytics** services. The samples can be found under 'sample.zementis' namespace prefix.

- ▲ WmPredictiveAnalytics
  - ▲ pub
    - ▲ zementis
      - ▲ apply
        - batchApply
        - recordApply
        - Record
      - ▲ model
        - deleteAllModels
        - deleteModel
        - getModelInfo
        - getModelSource
        - listModels
        - modelExists
        - uploadModel
        - Field
        - ModelInfo
      - ▲ resource
        - deleteAllResources
        - deleteResource
        - getResourceFile
        - getResourceInfo
        - listResources
        - resourceFileExists
        - uploadResource
        - ResourceInfo
    - ▲ sample
      - ▲ zementis
        - batchApply
        - batchApplyWithResources
        - pmmItoDocument
        - singleRecordApply
        - updateModel
        - updateResource
  - ▲ wm
    - ▲ zementis
      - ▲ config
        - getConfig
        - setConfig
        - shutDown
        - startUp
      - ▲ model
        - activateModel
        - deactivateModel
        - getSerializedModel



## 3.2 Model

Perform operations on predictive models.

### 3.2.1 *pub.zementis.model:Field*

Field properties document

#### *Fields*

- *name* **String** - field name
- *type* **String** - field data type with values: *STRING*, *BOOLEAN*, *INTEGER*, *FLOAT*, *DOUBLE*, *DATE*, *DATETIME*, or *TIME*
- *usage* **String** - field usage with values: *ACTIVE*, *SUPPLEMENTARY*, *TARGET*, *GROUP*, *DERIVED*, or *OUTPUT*

### 3.2.2 *pub.zementis.model:ModelInfo*

Document with model properties

#### *Fields*

- *modelName* **String** - model name
- *description* **String** - description of the model from PMML
- *isActive* **Boolean** - true if model is active and ready for data to be applied, false otherwise
- *inputFields* [pub.zementis.model:Field](#) **List** - list of model input fields as **IData** objects
- *outputFields* [pub.zementis.model:Field](#) **List** - list of model output fields as **IData** objects

### 3.2.3 *pub.zementis.model:listModels*

List all existing model names.

#### *Input Parameters*

None

#### *Output Parameters*

- *models* **String List** - list of existing model names

### 3.2.4 *pub.zementis.model:modelExists*

Test for existence of specified model name.

#### *Input Parameters*

- *modelName* **String** - model name

#### *Output Parameters*

- **exists** **Boolean** - true if model exists, false otherwise

### 3.2.5 *pub.zementis.model:getModelInfo*

Get model properties for specified model name.

#### *Input Parameters*

- *modelName* **String** - model name

#### *Output Parameters*

- *modelInfo* [pub.zementis.model:ModelInfo](#) - model properties as **IData** object

### 3.2.6 *pub.zementis.model:uploadModel*

Upload new **PMML** model.

#### *Input Parameters*

- *model* **Document** - PMML model source choice as **IData** object
  - *string* **String** - PMML model as string
  - *stream* **java.io.InputStream** - PMML model as input stream object
  - *filePath* **String** - file path to PMML model file
- *defaultModelName* **String** - default model name in case optional **PMML** *modelName* attribute is empty
- *applyCleanser* **Boolean** Optional - if true, performs comprehensive syntactic and semantics checks and corrections on the uploaded PMML, default value: true

#### *Output Parameters*

- *modelInfo* [pub.zementis.model:ModelInfo](#) - new model properties as **IData** object

### 3.2.7 *pub.zementis.model:getModelSource*

Get **PMML** model source for specified model.

#### *Input Parameters*

- *modelName* **String** - model name
- *annotated* **Boolean** Optional - if true, returns **PMML** source with annotations, default value: false

#### *Output Parameters*

- *modelSource* **String** - PMML model source

### 3.2.8 *pub.zementis.model:deleteModel*

Delete model for specified model name.

#### *Input Parameters*

- *modelName* **String** - model name

#### *Output Parameters*

- *models* **String List** - list of remaining model names

### 3.2.9 *pub.zementis.model:deleteAllModels*

Delete all existing models.

#### *Input Parameters*

None

#### *Output Parameters*

- *models* **String List** - list of remaining model names

### 3.2.10 *wm.zementis.model:activateModel*

Activate model for specified model name.

#### *Input Parameters*

- *modelName* **String** - model name

#### *Output Parameters*

- *modelInfo* [pub.zementis.model:ModelInfo](#) - updated model properties as **IData** object

### 3.2.11 *wm.zementis.model:deactivateModel*

Deactivate model for specified model name.

#### *Input Parameters*

- *modelName* **String** - model name

#### *Output Parameters*

- *modelInfo* [pub.zementis.model:ModelInfo](#) - updated model properties as **IData** object

### 3.2.12 *wm.zementis.model:getSerializedModel*

Get model as serialized binary file for specified model name.

### *Input Parameters*

- *modelName* **String** - model name

### *Output Parameters*

- *serializedModelInputStream* **java.io.InputStream** - serialized model as input stream object

## 3.3 Apply

Apply predictive models to data.

### 3.3.1 *pub.zementis.apply:Record*

Map of field names as **String** and values as **Object** used as input and output data type for apply operations.

#### *Fields*

Any **String/Object** pair

### 3.3.2 *pub.zementis.apply:recordApply*

Apply the specified model to single record.

#### *Input Parameters*

- *modelName* **String** - model name
- *inputRecord* [pub.zementis.apply:Record](#) - input Record document as **IData** object

#### *Output Parameters*

- *outputRecord* [pub.zementis.apply:Record](#) - output Record document as **IData** object

### 3.3.3 *pub.zementis.model:batchApply*

Apply specified model to multiple input records.

#### *Input Parameters*

- *modelName* **String** - model name
- *data* **Document** - input data type selector as **IData** objects
  - *records* [pub.zementis.apply:Record](#) **List** - list of Record documents as **IData** objects
  - *stream* **java.io.InputStream** - CSV or JSON data file input stream
  - *filePath* **String** - file path to CSV or JSON data file

#### *Output Parameters*

- *outputRecords* [pub.zementis.apply:Record](#) List - list of Record documents as **IData** objects

## 3.4 Resource

Manage model resources. Predictive models may require external resources such as custom functions, look-up tables or training data tables.

### 3.4.1 *pub.zementis.resource:ResourceInfo*

Resource file properties

#### *Fields*

- *fileName* **String** - resource file name used to identify resource
- *resourceType* **String** - type of resource
- *resourceIdentifier* **String** - resource identifier used in PMML model
- *resourceNames* **String List** - list of custom function names or table names contained in this resource file

### 3.4.2 *pub.zementis.resource:listResources*

Get list of resource properties for all existing resource files

#### *Input Parameters*

None

#### *Output Parameters*

- *resourceInfoList* [pub.zementis.resource:ResourceInfo](#) List - list of all resources as **IData** objects

### 3.4.3 *pub.zementis.resource:resourceFileExists*

Test resource file existence for specified resource file name.

#### *Input Parameters*

- *resourceFileName* **String** - resource file name

#### *Output Parameters*

- *exists* **Boolean** - true if resource file exists, false otherwise

### 3.4.4 *pub.zementis.resource:getResourceInfo*

Get resource properties for specified resource file name.

#### *Input Parameters*

- *resourceFileName* **String** - resource file name

### Output Parameters

- *resourceInfo* [pub.zementis.resource:ResourceInfo](#) - resource file properties as **IData** object

### 3.4.5 *pub.zementis.resource:uploadResource*

Upload new resource file.

#### Input Parameters

- *resourceFileName* **String** - resource file name
- *resourceFile* **Document** - choice of resource file input type as **IData** object
  - *filePath* **String** - file path to resource file
  - *stream* **java.io.InputStream** - resource file as stream object
  - *bytes* **byte[]** - resource file as byte array

#### Output Parameters

- *resourceInfo* [pub.zementis.resource:ResourceInfo](#) - new resource properties as **IData** object

### 3.4.6 *pub.zementis.resource:getResourceFile*

Get resource file for specified resource file name.

#### Input Parameters

- *resourceFileName* **String** - resource file name
- *getResourceAs* **String** - Select returned resource file data type: *bytes* or *stream*

#### Output Parameters

- *resourceFile* **Document** - returned resource file data type as **IData** object
  - *bytes* **byte[]** - resource file as byte array
  - *stream* **java.io.InputStream** - resource file as stream object

### 3.4.7 *pub.zementis.resource:deleteResource*

Delete resource for specified resource file name.

#### Input Parameters

- *resourceFileName* **String** - resource file name

#### Output Parameters

- *resourceInfoList* [pub.zementis.resource:ResourceInfo](#) **List** - list of all remaining resources as **IData** objects

### 3.4.8 *pub.zementis.resource:deleteAllResources*

Delete all existing resources.

#### *Input Parameters*

None

#### *Output Parameters*

- *resourceInfoList* [pub.zementis.resource:ResourceInfo](#) List - list of all remaining resources as IDa-ta objects

## 3.5 Configuration

Configure `WmPredictiveAnalytics` package.

### 3.5.1 *wm.zementis.config:getConfig*

Get package configuration properties.

#### *Input Parameters*

None

#### *Output Parameters*

- *hostUrl* String - Zementis Server host URL
- *username* String - Zementis Server user name

### 3.5.2 *wm.zementis.config:setConfig*

Set package configuration properties.

#### *Input Parameters*

- *hostUrl* String - Zementis Server host URL
- *username* String - Zementis Server user name
- *password* String - Zementis Server user password

#### *Output Parameters*

None

### 3.5.3 *wm.zementis.config:testConnection*

Test connection to Zementis Server without saving configuration.

#### *Input Parameters*

- *hostUrl* String - Zementis Server host URL
- *username* String - Zementis Server user name
- *password* String - Zementis Server user password

#### *Output Parameters*

None

### **3.5.4 *wm.zementis.config:startup***

Activate and initialize package.

#### *Input Parameters*

None

#### *Output Parameters*

None

### **3.5.5 *wm.zementis.config:shutdown***

Safely deactivate package.

#### *Input Parameters*

None

#### *Output Parameters*

None

## **3.6 Samples**

Flow service examples using public services.

### **3.6.1 *sample.zementis:singleRecordApply***

Upload Iris PMML model and score single record.

### **3.6.2 *sample.zementis:batchApply***

Upload arbitrary PMML model and perform batch scoring with data file.

### **3.6.3 *sample.zementis:batchApplyWithResources***

Upload arbitrary PMML model with two resource files and perform batch scoring with data file.



### ***3.6.4 sample.zementis:updateModel***

Update arbitrary PMML model.

### ***3.6.5 sample.zementis:updateResource***

Update arbitrary resource file.

### ***3.6.6 sample.zementis:pmmlToDocument***

Upload PMML model and convert it to document.