

# **Apama Predictive Analytics Add-on**

Innovation Release

Version 10.0

April 2017

This document applies to Apama Predictive Analytics Add-on Version 10.0 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 © 2013-2017 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).

Use, reproduction, transfer, publication or disclosure is prohibited except as specifically provided for in your License Agreement with Software AG.

# Table of Contents

<b>About this Guide.....</b>	<b>5</b>
Documentation roadmap.....	5
Online Information.....	5
Contacting customer support.....	6
<b>Release Notes.....</b>	<b>7</b>
What's new in 10.0 release.....	8
What's new in 9.12 release.....	8
What's new in 9.10 release.....	8
<b>Introduction to Predictive Analytics Add-on.....</b>	<b>11</b>
Predictive Analytics Engine.....	12
Predictive Analytics Plug-in.....	12
<b>Installing Predictive Analytics Add-on.....</b>	<b>13</b>
Getting Started.....	14
<b>Working with Sample Project.....</b>	<b>15</b>
Running the sample project in Software AG Designer.....	16
Running the sample project in Apama command prompt.....	16
Running the sample project using ant configuration.....	17
Sample output.....	17
<b>Working with Predictive Analytics Plug-in.....</b>	<b>19</b>
User application EPL.....	20
Adding, updating, and removing PMML models.....	21
Loading custom resources.....	22
Injecting the Predictive Analytics Plug-in using Ant Macro.....	23
Connecting Predictive Analytics Plug-in to Messaging Services.....	24
Connecting Predictive Analytics Plug-in to Digital Event Services.....	25



## About this Guide

---

This guide describes how to install and configure the Apama Predictive Analytics Add-on.

## Documentation roadmap

---

Predictive Analytics Plug-in provides documentation in the following formats:

- HTML (viewable in a web browser)
- PDF (available from the documentation website)

You can access the HTML documentation on your machine after Predictive Analytics Plug-in has been installed:

- **Windows.** Select **Start > All Programs > Software AG > Tools > Predictive Analytics Add-on *n.n* > Predictive Analytics Plug-in Documentation *n.n***. Note that **Software AG** is the default group name that can be changed during the installation.

Predictive Analytics Plug-in also provides the following API reference information:

- API Reference for Predictive Analytics Plugin EPL (Apamadoc)
- API Reference for Predictive Analytics Engine (Javadoc)

## Online Information

---

### Software AG Documentation Website

You can find documentation on the Software AG Documentation website at <http://documentation.softwareag.com>. The site requires Empower credentials. If you do not have Empower credentials, you must use the TECHcommunity website.

### Software AG Empower Product Support Website

You can find product information on the Software AG Empower Product Support website at <https://empower.softwareag.com>.

To submit feature/enhancement requests, get information about product availability, and download products, go to [Products](#).

To get information about fixes and to read early warnings, technical papers, and knowledge base articles, go to the [Knowledge Center](#).

### Software AG TECHcommunity

You can find documentation and other technical information on the Software AG TECHcommunity website at <http://techcommunity.softwareag.com>. You can:

- Access product documentation, if you have TECHcommunity credentials. If you do not, you will need to register and specify "Documentation" as an area of interest.
- Access articles, code samples, demos, and tutorials.
- Use the online discussion forums, moderated by Software AG professionals, to ask questions, discuss best practices, and learn how other customers are using Software AG technology.
- Link to external websites that discuss open standards and web technology.

## Contacting customer support

---

If you have an account, you may open Apama Support Incidents online via the eService section of Empower at <https://empower.softwareag.com/>. If you do not yet have an account, send an email to [empower@softwareag.com](mailto:empower@softwareag.com) with your name, company, and company email address and request an account.

If you have any questions, you can find a local or toll-free number for your country in our Global Support Contact Directory at [https://empower.softwareag.com/public\\_directory.asp](https://empower.softwareag.com/public_directory.asp) and give us a call.

# 1 Release Notes

■ What's new in 10.0 release .....	8
■ What's new in 9.12 release .....	8
■ What's new in 9.10 release .....	8

Release Notes describes the changes introduced with the current Apama Predictive Analytics Add-on release as well as earlier releases.

## What's new in 10.0 release

---

- Predictive Analytics Engine has been updated to ADAPA 5.0.1. If you have an existing installation, you must back up the artifacts uploaded in Predictive Analytics Engine before you upgrade. Upgrading to ADAPA 5.0.1 requires a clean repository store. The following features are available with this upgrade :
  - Compatibility and automatic model conversion to PMML 4.3
  - Support for new algorithms: Convolutional Neural Networks and k-Nearest Neighbor
- Added support for adding and updating models received as byte stream to Predictive Analytics plug-in.

## What's new in 9.12 release

---

- Predictive Analytics Engine has been migrated from ADAPA 4.1 to 4.2.4.
- Added new ModelManager API to add, remove, update PMML models. Refer to ModelManager API in "API Reference for Predictive Analytics Plug-in EPL (ApamaDoc)".
- Added new ResourceManager API to add or remove resources like custom functions and lookup tables. Refer to ResourceManager API in "API Reference for Predictive Analytics Plug-in EPL (ApamaDoc)".
- Added docker packaging kit for Predictive Analytics Add-on (this is available only on Linux).
- Samples are moved from APAMA\_HOME/adapters/samples to APAMA\_HOME/samples/PredictiveAnalytics.
- Added samples for demonstrating integration with ModelManager API, ResourceManager API, Integration with various kinds of Asset Stores, and building a docker enabled application.
- Added an installer dialog for providing license file for Predictive Analytics Engine. The license file can also be copied to APAMA\_WORK/license directory after installation.

## What's new in 9.10 release

---

- New service parameters are added to Predictive Analytics Add-on



- **subscribeToChannel.** Applications can send input scoring requests to this channel for consumption by the Predictive Analytics plug-in.
- **sendToChannel.** Predictive analytics plug-in will publish the output predictions to this channel. User applications can subscribe to this channel for receiving responses.
- **maxBatchSize.** To set the maximum number of input events that will be grouped together and consumed by the plug-in. Default is set to 1000.



## 2 Introduction to Predictive Analytics Add-on

---

■ Predictive Analytics Engine .....	12
■ Predictive Analytics Plug-in .....	12

The Predictive Analytics Add-on includes the following components:

- Predictive Analytics Engine
- Predictive Analytics Plug-in

Predictive Analytics Plug-in combines Zementis ADAPA's predictive model deployment and scoring capabilities with Apama's comprehensive big data analytics platform to create an integrated solution. Predictive Analytics Plug-in encloses ADAPA library that allows you to set up a context and score data.

## Predictive Analytics Engine

---

Predictive Analytics Engine uses the ADAPA scoring engine from Zementis to deliver a fast, standards-based deployment platform for predictive analytics. Some of the salient features of Zementis ADAPA include:

- Scoring engine (decision engine) for predictive analytics
  - Uses predictive models to score data
  - Delivers precise insights into market dynamics, security risks and sensor information
  - Facilitates informed decision making based on quantitative logic and insights
- Standards-based platform for deploying predictive analytics
  - Employs the Predictive Model Markup Language (PMML) standard to import and deploy predictive models
  - Allows data science teams to rapidly transform analytical models into operational tools for business users
  - Allows organizations to rapidly deploy, run and manage predictive models that enable real-time insights and actions for the business

## Predictive Analytics Plug-in

---

The predictive analytics plug-in is an Apama correlator plug-in for integrating with Zementis ADAPA to score predictive models from within Apama applications. For more information on APIs used by Predictive Analytics Plug-in, see "API Reference for Predictive Analytics Plug-in EPL (ApamaDoc)".

To get started with Predictive Analytics plug-in, see ["Getting Started" on page 14](#) and ["Working with Predictive Analytics Plug-in" on page 19](#).

# 3 Installing Predictive Analytics Add-on

---

■ Getting Started ..... 14

To get started with the installation, see *Installing Software AG Products* guide. It is intended for use with the following guides:

- *Using the Software AG Installer*. This guide explains how to prepare your machine to use the Software AG Installer, and how to use the Software AG Installer and Software AG Uninstaller to install and uninstall your products.
- *Using the Software AG Update Manager*. This guide explains how to use the Software AG Update Manager to install and uninstall fixes on your Software AG products.
- *Upgrading Software AG Products*. This guide contains information on how to upgrade Apama.

The most up-to-date versions of these guides are always available at <http://documentation.softwareag.com/> (Empower login required).

**Note:** When you are installing Predictive Analytics Add-on using Software AG installer, you are prompted for the Predictive Analytics Engine license file. Optionally, if you specify the license file during the installation process, the license file will be copied to the APAMA\_WORK/license directory. You can also copy the license file to the APAMA\_WORK/license directory after the installation process. The license file is mandatory to start the Predictive Analytics Engine.

## Getting Started

---

The Predictive Analytics plug-in ships with a set of documentation files. All the relevant documentation for Predictive Analytics Plugin can be found in APAMA\_HOME/adapters/doc folder.

## 4 Working with Sample Project

---

■ Running the sample project in Software AG Designer .....	16
■ Running the sample project in Apama command prompt .....	16
■ Running the sample project using ant configuration .....	17
■ Sample output .....	17

The Predictive Analytics plug-in package ships a sample project **EnergyData** available at `APAMA_HOME/samples/PredictiveAnalytics/EnergyData`. You can use this as a reference for using Predictive Analytics Plug-in.

It is recommended that you copy this sample folder to `APAMA_WORK` directory rather than running it directly from the installation directory. For Windows users with UAC enabled, this step is required to avoid access denied errors when writing to the sample directory.

Ensure that the Predictive Analytics Engine license file (`zementis.license`) is copied to `APAMA_WORK/license` folder.

Running the sample project performs the following tasks:

- Starts correlator, injects the plugin bundle, initialises the Predictive Analytics plug-in by injecting `EnergyDataSample.mon` and the input is sent from `EnergyData.evt`.
- Apama EPL application `EnergyDataSample.mon` performs the following tasks to configure the Predictive Analytics plug-in:
  - Creates a new `ServiceParams` instance and provides details about the PMML model to be loaded.
  - Uses a `ServiceHandlerFactory` to create a new Predictive Analytics plug-in instance with the configured parameters.
  - Receives a callback on `onServiceInitialised` with reference to the newly created `ServiceHandler` after the Predictive Analytics plug-in instance is successfully created.
  - Forwards any requests (`SampleInput`) received to the Predictive Analytics engine by creating a new `com.apama.pa.pmml.Input` request.
  - Consumes responses (`com.apama.pa.pmml.Output`) from the Predictive Analytics engine.

---

## Running the sample project in Software AG Designer

To run the sample project in Software AG Designer, see *Using Apama with Software AG Designer*.

---

## Running the sample project in Apama command prompt

Before you can run any of the Apama tools, you must set up the environment correctly. See "Setting up the environment using the Apama Command Prompt" in *Deploying and Managing Apama Applications*.

---

### To run the sample project in Apama command prompt

1. Change to directory to the location where the sample project is located.



2. Start the correlator with Java support enabled:

```
correlator --java
```

3. Inject the required monitors:

```
engine_inject --java
"$APAMA_HOME/adapters/lib/Predictive-Analytics-Plugin.jar"
engine_inject --cdp
"$APAMA_HOME/adapters/monitors/predictive_analytics_plugin_monitors.cdp"
```

4. Inject the MonitorScript file to run the sample:

```
engine_inject "monitors/EnergyDataSample.mon"
```

5. Send the EPL application as input to the sample:

```
engine_send "events/EnergyData.evt"
```

## Running the sample project using ant configuration

You can use the ant configuration to run the sample project in Apama command prompt on Windows and UNIX platforms. Before you can run any of the Apama tools, you must set up the environment correctly. See "Setting up the environment using the Apama Command Prompt" in *Deploying and Managing Apama Applications*.

### To run the sample project using ant configuration

1. Change to APAMA\_HOME/samples/PredictiveAnalytics/EnergyData directory.
2. Start the sample project by using the command `ant start`.

This command performs the following tasks:

- Starts the correlator with Java support.
- Injects the plug-in jar file and its associated `predictive_analytics_plugin_monitors.cdp` package to the correlator.
- Injects the EnergyData sample monitor to the correlator available at `APAMA_HOME/samples/PredictiveAnalytics/EnergyData/monitors/EnergyDataSample.mon`
- Initializes the plug-in to load the PMML file from `APAMA_HOME/samples/PredictiveAnalytics/EnergyData/model/EnergyDataModel.pmml`
- Sends sample prediction data.

3. Stop the sample project by using the command `ant stop`.

## Sample output

The correlator log will show messages similar to the following:

```
com.apama.pa.pmml.sample.PredictiveAnalytics_EnergyData_Sample
[1] {"Predicted_Usage":"16.18362364781374"}
```

```
com.apama.pa.pmml.sample.PredictiveAnalytics_EnergyData_Sample  
[1] {"Predicted_Usage":"15.397684338406936"}  
com.apama.pa.pmml.sample.PredictiveAnalytics_EnergyData_Sample  
[1] {"Predicted_Usage":"19.12970126490951"}  
com.apama.pa.pmml.sample.PredictiveAnalytics_EnergyData_Sample  
[1] {"Predicted_Usage":"15.796460465819097"}  
com.apama.pa.pmml.sample.PredictiveAnalytics_EnergyData_Sample  
[1] {"Predicted_Usage":"21.046370444450062"}  
com.apama.pa.pmml.sample.PredictiveAnalytics_EnergyData_Sample  
[1] {"ADAPA_Error":"Value [NA] is invalid for field [PreUse]."}  

```

Errors encountered while scoring requests are sent by populating `ADAPA_Error` field in `com.apama.pa.pmml.Output` event.

# 5 Working with Predictive Analytics Plug-in

---

■ User application EPL .....	20
■ Adding, updating, and removing PMML models .....	21
■ Loading custom resources .....	22
■ Injecting the Predictive Analytics Plug-in using Ant Macro .....	23
■ Connecting Predictive Analytics Plug-in to Messaging Services .....	24
■ Connecting Predictive Analytics Plug-in to Digital Event Services .....	25

Ensure that you have set the correlator classpath to the Zementis ADAPA license directory.

### To start the plug-in

1. Start the correlator.
2. Inject Predictive-Analytics-Plugin.jar located at APAMA\_HOME/adapters/lib to the correlator.
3. Inject predictive\_analytics\_plugin\_monitors.cdp CDP file located at APAMA\_HOME/adapters/monitors to the correlator.
4. Inject user application EPL.

## User application EPL

A user application EPL script should perform the following tasks:

1. Create an instance of ServiceParams.

```
com.apama.pa.pmml.ServiceParams serviceParams :=
com.apama.pa.pmml.ServiceParamsHelper.create();
```

2. Set the configuration parameters.

```
serviceParams.setPMMLFileName("PMML_CONFIG_FILE_NAME");
serviceParams.addResource(CUSTOM_RESOURCE_NAME1);
serviceParams.addResource(CUSTOM_RESOURCE_NAME2);
```

For a full list of configuration parameters, see "API Reference for Predictive Analytics Plug-in EPL (ApamaDoc)".

3. Request the ServiceHandlerFactory to create a new service handler and pass the ServiceParams.

```
com.apama.pa.pmml.ServiceHandlerFactory
.create(com.apama.pa.pmml.ServiceName.Zementis,
"PREDICTIVE_ANALYTICS_INSTANCE_1",
serviceParams,
onServiceInitialised,
onServiceError);
```

You must pass two additional callbacks to the service handler factory.

- This callback is called when the PMML file is successfully loaded and the service is initialised. The ServiceHandler received in this callback can be used to retrieve the list of models available for this service.

```
action onServiceInitialised(com.apama.pa.pmml.ServiceHandler
servicehandler) {
    //Implement your application logic here
}
```

- This callback is called if an error is encountered while loading the PMML file or when there is an issue with the input.

```
action onServiceError(com.apama.pa.pmml.ServiceError serviceError) {
    log "Received Service Error " + serviceError.getErrorMessage() at ERROR;
```

```
}
```

#### 4. Create an input event and pass it to the plug-in.

```
com.apama.pa.pmml.Input input := new com.apama.pa.pmml.Input;
input.instanceName := "PREDICTIVE_ANALYTICS_INSTANCE_1";
input.modelName := "SAMPLE_MODEL_NAME";
input.requestId := integer.getUnique().toString();
input.inputFields.add("FIELD_1", "FIELD_1_VALUE");
input.inputFields.add("FIELD_2", "FIELD_2_VALUE");
input.inputFields.add("FIELD_2", "FIELD_3_VALUE");
route input;
```

#### 5. Check for the output event which corresponds to the specified input.

```
com.apama.pa.pmml.Output output;
on all com.apama.pa.pmml.Output
(instanceName="PREDICTIVE_ANALYTICS_INSTANCE_1") : output
{
  log output.toString() at INFO;
  //Do additional processing
}
```

Error handling when processing an input request:

- If there is a significant error while processing the input request, you will receive a callback on the `onServiceError` callback registered during service initialisation.
- Errors and warnings reported by the Predictive Analytics Engine are also propagated through the output event.
  - If any errors are found during scoring, search for `ADAPA_Error` in the `outputFields`

Example:

```
com.apama.pa.pmml.Output("Instance_1","206",
  {"ADAPA_Error":"Value [NA] is invalid for field [PreUse]."},{})
```

- Warnings reported by the scoring engine are also forwarded in the `outputFields` as `ADAPA_Warning_<N>`, where N can be 1, 2, 3 ...

Example:

```
com.apama.pa.pmml.Output("Instance_1","206",
  {"ADAPA_Warning_1":"warning message",
   "Predicted_Usage":"19.980840445004088"},{})
```

## Adding, updating, and removing PMML models

The Predictive Analytics Add-on supports adding, updating, and removing a PMML model at runtime.

### To add, update, or remove a model in an EPL script

1. The Predictive Analytics Add-on also supports adding and removing resources at runtime through ModelManager API. ModelManager API can be accessed by calling `getModelManager()` on the `ServiceHandler` object received in service initialisation `onServiceInitialised` callback.

```

action onServiceInitialised(ServiceHandler servicehandler)
{
    ModelManager modelmanager := serviceHandler.getModelManager();
    //Add Model1 from PMML_PATH1
    modelmanager.addModel("PMML_PATH1");
    //Add Model2 from PMML_PATH2
    modelmanager.addModel("PMML_PATH2");
    ...
    ...
    //Update Model1 from another source PMML_PATH3
    modelmanager.updateModel("Model1", "PMML_PATH3");
    ...
    ...
    //remove model when done
    modelmanager.removeModel("Model1");
    modelmanager.removeModel("Model2");
}

```

Any errors are reported through default callback `onServiceError` of service handler.

You can add custom callbacks for the above mentioned functions as described below:

```

action onServiceInitialised(ServiceHandler servicehandler)
{
    ModelManager modelmanager := serviceHandler.getModelManager();
    //Add Model1 from PMML_PATH1
    modelmanager.addModelCb("PMML_PATH1", onStatus);
    //Add Model2 from PMML_PATH2
    modelmanager.addModelCb("PMML_PATH2", onStatus);
    ...
    ...
    //Update Model1 from another source PMML_PATH3
    modelmanager.updateModelCb("Model1", "PMML_PATH3", onStatus);
    ...
    ...
    //remove model when done
    modelmanager.removeModelCb("Model1", onStatus);
    modelmanager.removeModelCb("Model2", onStatus);
}
action onStatus(ServiceError serviceError)
{
    log "Received status on configured callback:
    "+serviceError.getErrorMessage();
}

```

For more information, see "API Reference for Predictive Analytics Plug-in EPL (ApamaDoc)".

For more information, refer to the samples at `APAMA_HOME/samples/PredictiveAnalytics/EnergyData ModelManager`

## Loading custom resources

The Predictive Analytics Add-on supports adding and removing custom resources like lookup tables and custom functions. You can follow these steps to load and use custom functions in EPL.

### To add or remove custom resources in an EPL script

1. Create an instance of `ServiceParams`.

```
com.apama.pa.pmml.ServiceParams serviceParams :=
    (new com.apama.pa.pmml.ServiceParamsHelper).create();
```

2. Add any custom resource either with absolute path or relative path.

```
serviceParams.addResource(CUSTOM_RESOURCE_NAME1);
serviceParams.addResource(CUSTOM_RESOURCE_NAME2);
```

The Predictive Analytics Add-on also supports adding and removing resources at runtime through `ResourceManager` API. `ResourceManager` API can be accessed by calling `getResourceManager()` on the `ServiceHandler` object received in service initialization `onServiceInitialised` callback.

```
action onServiceInitialised(ServiceHandler servicehandler)
{
    serviceHandler := servicehandler;
    ResourceManager resourcemanager := serviceHandler.getResourceManager();
    resourcemanager.addResource("CUSTOM_RESOURCE_NAME1");
    resourcemanager.addResource("CUSTOM_RESOURCE_NAME2");
    resourcemanager.addResource("CUSTOM_RESOURCE_NAME3");
    ModelManager modelmanager := serviceHandler.getModelManager();
    //Add MODEL_NAME1 from PMML_PATH1
    modelmanager.addModel("PMML_PATH1");
    //Add MODEL_NAME2 from PMML_PATH2
    modelmanager.addModel("PMML_PATH2");
    //application code ...
    //remove model when done
    modelmanager.removeModel("MODEL_NAME1");
    modelmanager.removeModel("MODEL_NAME1");
    //remove resource when done
    resourcemanager.removeResource("CUSTOM_RESOURCE_NAME1");
    resourcemanager.removeResource("CUSTOM_RESOURCE_NAME2");
    resourcemanager.removeResource("CUSTOM_RESOURCE_NAME3");
}
```

You can also list the resources that are added by calling `listResources()` function using `ResourceManager` object. The resources are removed automatically when the engine stops, but it is recommended to explicitly remove the unused resources.

For more information, refer to the samples at:

- `APAMA_HOME/samples/PredictiveAnalytics/ECommerceFraud Custom Functions`
- `APAMA_HOME/samples/PredictiveAnalytics/ECommerceFraud Custom Context`

## Injecting the Predictive Analytics Plug-in using Ant Macro

You can inject the Predictive Analytics package in to the correlator using the ant macro file.

### To inject the Predictive Analytics plug-in in to the correlator

1. Import the ant macro file `${APAMA_HOME}\adapters\ant_macros\predictive-analytics-support-macros.xml` to the ant user script.
2. Add the dependency on 'predictive-analytics-plugin-bundle' ant target to inject the Predictive Analytics add-on components in to the correlator.

## Connecting Predictive Analytics Plug-in to Messaging Services

Predictive Analytics plug-in can accept and process PMML data received from messaging services like Digital Event Services, MQTT, Universal Messaging and so on.

### Working with PMML data received from messaging services

The Predictive Analytics plug-in supports data received from other messaging services through interfaces `addModelFromStream`, `addModelFromStreamCb`, `updateModelFromStream`, `updateModelFromStreamCb`. For more information on interfaces, see [ApamaDoc](#).

```
action onServiceInitialised(ServiceHandler servicehandler)
{
    ModelManager modelmanager := servicehandler.getModelManager();
    // Add MODEL_NAME of PMML_DATA received as sequence of integers
    // any errors while loading a model will be reported
    // using the default error callback registered during service initialisation
    modelmanager.addModelFromStream(MODEL_NAME, PMML_DATA);
    // Alternatively register a new error handler
    //modelmanager.addModelFromStreamCb(MODEL_NAME, PMML_DATA, modelConfigError);
    ..
    ..
    // Update an exiting PMML model with another source pmmlData2 received as
    // sequence of integers, any errors while updating the model will be reported
    // using the default error callback registered during service initialisation
    modelmanager.updateModelFromStream(MODEL_NAME, UPDATE_PMML_DATA);
    // Alternatively register a new error handler
    //modelmanager.updateModelFromStreamCb
    (MODEL_NAME, UPDATE_PMML_DATA, modelConfigError);
    ...
    ...
}
// Optional error handler for reeporting errors encountered while setting up PMML models
action modelConfigError(ServiceError serviceError)
{
    log "Model Manager Stream Sample, error with Add/Update model" +
        serviceError.getErrorMessage() at ERROR;
}
// Error handler for reeporting errors encountered with the instance
action onServiceError(ServiceError serviceError)
{
    log "Model Manager Stream Sample " + serviceError.getErrorMessage()
        at ERROR;
}
```



## Connecting Predictive Analytics Plug-in to Digital Event Services

1. Start Universal Messaging server.
2. Create a digital event definition for sending a PMML model using digital event service.
3. Define a byte array field in the event definition to carry PMML data. You can optionally define additional fields to store information like modelName, add/update/delete model, instanceName and so on.
4. Place the created event in APAMA\_HOME/common/DigitalEventServices/TypeRepository.

You must configure the Predictive Analytics plug-in to connect to digital event service.

### To configure the Predictive Analytics plug-in

1. Create a new Apama project. In the **New Apama Project wizard**, select the **Digital Event Services** bundle and **Predictive Analytics Plug-in** bundle.
2. In the project's **config > connectivity > DigitalEventServices** node, double click **EventTypeList.apamades**.
3. Select the digital event types to convert to Apama event definitions. Edit and save the EventTypeList.apamades file.

Software AG Designer automatically generates the EPL files containing the Apama event definitions for the selected event types in the project's **autogenerated > DigitalEventTypes** node.

4. In the project's **config > connectivity > DigitalEventServices**, double click DESConnectivity.properties file and edit the property DigitalEventServices\_replaceConfigWithRNAME with realm URL to connect to Universal messaging.
5. In the sample project's monitors, LoadPmmlFromStreamSample.mon subscribes for channel DES\_MODEL\_EVENT and opens an event listener. Here addModelFromStream of modelManager is called to process received pmmlData.

```
using com.softwareag.connectivity.ConnectivityPlugins;
using DES_MODEL_EVENT;
...
ConnectivityPlugins.onApplicationInitialized();
DES_MODEL_EVENT.subscribe(); // This context will now receive digital events
// of type 'DES_MODEL_EVENT'
action onServiceInitialised(ServiceHandler servicehandler)
{
    ModelManager modelmanager := serviceHandler.getModelManager();
    DES_MODEL_EVENT modelEvt;
    on all DES_MODEL_EVENT(instanceName=config.instanceName) :modelEvt
    {
        modelManager.addModelFromStream(MODEL_NAME,modelEvt.pmmlData);
    }
}
```

```
}  
}
```

6. Run the Apama project.