

# Deltek Maconomy 2.3 GA

## MOL Language Reference

**December 2, 2016**

While Deltek has attempted to verify that the information in this document is accurate and complete, some typographical or technical errors may exist. The recipient of this document is solely responsible for all decisions relating to or use of the information provided herein.

The information contained in this publication is effective as of the publication date below and is subject to change without notice.

This publication contains proprietary information that is protected by copyright. All rights are reserved. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, or translated into another language, without the prior written consent of Deltek, Inc.

This edition published December 2016.

© Deltek, Inc.

Deltek's software is also protected by copyright law and constitutes valuable confidential and proprietary information of Deltek, Inc. and its licensors. The Deltek software, and all related documentation, is provided for use only in accordance with the terms of the license agreement. Unauthorized reproduction or distribution of the program or any portion thereof could result in severe civil or criminal penalties.

All trademarks are the property of their respective owners.

# Contents

Introduction ..... 1

Language Definition ..... 2

    Object ..... 2

    Field Definitions ..... 3

    Maconomy Types ..... 3

## Introduction

Maconomy Object Language (MOL) is a language for defining custom database objects to be integrated in a standard Maconomy installation. Using MOL, new database objects are defined in Maconomy terminology.

This documentation reference describes the MOL language syntax.



For further information about the installation and use of MOL objects, see *Deltek Maconomy MBuilder Reference* and *Deltek Maconomy MUL Language Reference* documentation.

---

# Language Definition

This section describes the current version of MOL. The formal syntax of MOL is presented in BNF (Bachus Naur Form). The syntax of MOL is tag-based, and it consists of elements and attributes just like XML. Unlike XML, every attribute in MOL has an associated type, and can have a short form.

Note that MOL is not case-sensitive.

```
mol ::=
  <MOL 1.0>
  object
```

## Object

The core element of MOL is the object element. Exactly one instance of the object element must be defined. Inside the object element are field definitions, and so on.

```
object ::=
  <object (name = id | id) [(title = string | string)] >
  fieldDefinition+
  <end object>
```

Attributes of the object element are described in the following table.

Attribute	Description
name	The unique name identifier of the object. The name must be prefixed by a three-letter namespace followed by '_', for example, "abc_MySalary" where "abc" is the namespace.
[title]	The title of the object.

In the following example, a simple MOL object is defined. The namespace identifier is "abc", and the resulting object name is "abc\_MySalary."

```
<MOL 1.0>
<Object abc_MySalary>
  .Componentid :String :Key+
  .Target      :Integer
  .Salary      :Amount
  .SalaryGroup :String : "Salary level"
<End Object>
```

## Field Definitions

All field definitions reside inside the object element. A field consists of a field name, a field type, and possibly a field title and a key indicator attribute.

```
fieldDefinition ::=
  .id : (type = id | id) [:key(+ | -)]
  |
  (<field ((name = id | id) : (type = id) [:key(+ | -)]) >)
```

Attributes of the field element are as follows.

Attribute	Description
name	The name identifier of the field. The name must be unique within the object.
type	The type identifier attribute is mandatory. Available types are listed in “Maconomy Types”.
[key]	The optional key indicator attribute is used to indicate key fields of the object. The set of all key fields must identify the object entries uniquely.

In the previous example, the single field `Componentid` of type `String` is the only key field.

```
.Componentid :String :Key+
```

Thus, the field `Componentid` must uniquely identify the object entries.

## Maconomy Types

A set of predefined types in the Maconomy environment is available for use in MOL. Note that the database implementation of all Maconomy types is fixed and highly database-specific. See the “MBuilder Reference” documentation for further details on type conversions.

The types are divided into two groups: basic types and enumeration pop-up types.

### Basic Types

The basic types available from MOL are listed in the following table.

Type	Description
INTEGER	32-bit integer value
REAL	Floating point value
AMOUNT	Fixed Maconomy-style amount type
BOOLEAN	Boolean type
STRING	String type of max. 255 characters

Type	Description
DATE	Fixed Maconomy style date type
TIME	Fixed Maconomy style time type

## Popup Types

A large set of Maconomy version-specific enumeration types (“pop-up types”) are available in MOL. The set of available types and their corresponding values is Maconomy version-specific, and can be viewed using a Maconomy client in the Popup Fields dialog in the Setup module.

A blue geometric graphic consisting of several overlapping triangles and polygons, located in the top-left corner of the page.

Deltek is the leading global provider of enterprise software and information solutions for government contractors, professional services firms and other project- and people-based businesses. For decades, we have delivered actionable insight that empowers our customers to unlock their business potential. 20,000 organizations and millions of users in over 80 countries around the world rely on Deltek to research and identify opportunities, win new business, recruit and develop talent, optimize resources, streamline operations and deliver more profitable projects. Deltek – Know more. Do more.®

[deltek.com](http://deltek.com)