

Deltek Maconomy 2.3 GA

MCSL Quick Reference Guide

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

Overview	1
Introduction	2
Attributes and Referred Types	2
Document Structure	2
Namespace and Root Tag	5
Namespace	5
<MCSL>-Tag	5
Access Specification	6
Workspace Access Model	6
Outer <Access>-Tag	6
<Role>-Tag	6
Inner <Access>-Tag	6
<AllRoles>-Tag	7
<Workspace>-Tag	7
<Exclude>-Tag	7
Environment Specification	9
<Environment>-Tag	9
<Binding>-Tag	9
<Restriction>-Tag	10
<Condition>-Tag	10
Binding <Fields>-Tag	10
<Field>-Tag	10
<Define>-Tag	10
Define <Fields>-Tag	11
Reference <Fields>-Tag	11
Environment – Reserved Bindings	13
External Documentation Link	13
Menu Dock Decorations	13
Configuration Specification	16
<Configuration>-Tag	16
Dictionaries Specification	16
Update Sites Specification	16
<UpdateSite>-Tag	16
Pop-Ups Specification	17
<Popups>-Tag	17

<Popup>-Tag	18
-------------------	----

Overview

This document provides a quick reference guide to the Maconomy Coupling Service Language (MCSL).

Introduction

The MCSL is designed to enable the configuration of various configurable aspects of the Coupling Service. You can use MCSL to configure the following areas:

- Access control
- Environment holding contents of all Maconomy defined variables
- Dictionaries used for client localization
- Update sites locations

Attributes and Referred Types

Attributes of a given tag are described using the following table.

Attribute Name	Type	Usage
Attr1	Type of attr1	This attribute is used to...
Attr2	Type of attr2	Indicates...

In addition, attributes can have one of the following referred types.

Boolean	A Boolean Attribute (True or False)
Key	A string that is case-insensitive and never exposed to any end user. It is used for references (internally or from other parts in the spec/other specs).
Display	A string that is meant to be displayed to an end user (and which is therefore localized). It can never be used for any reference.
Id	A string that is case-insensitive and used to reference items in environments that are not controlled by Maconomy. It is never exposed to any end user.

Document Structure

An MCSL specification has the following basic structure:

```
<MCSL version="0.5"
xmlns="http://www.delttek.com/ns/mcsl">

  <Access>

    <Role source="All Workspaces">
      <Access all="true" />
    </Role>

    <Role source="TimeSheet User">
      <Access>
```

```

        <Workspace source="TimeSheet" />
    </Access>
</Role>
</Access>
<Environment>
    <Binding                                entity="UserInformation"
namespace="user:info">
        <Restriction condition="NameOfUser = userName()" />
        <Fields>
            <Field source="AccessToAllTimeSheets" />
            <Field source="CanSeeAllExpenseSheets" />
            <Field source="EmployeeNumber" />
            <Field source="NameOfUser" />
        </Fields>
    </Binding>
    <Binding                                entity="UserDialogGroup"
namespace="user:groups">
        <Restriction condition="NameOfUser = userName()" />
        <Fields>
            <Field source="GroupName" />
        </Fields>
    </Binding>
</Environment>
<Configuration>
    <UpdateSites>
        <UpdateSite    location="http://<someLocation>:1234/"
/>
        <UpdateSite    location="http://<otherLocation>:1234/"
/>
    </UpdateSites>
</Configuration>
</MCSL>

```

The document has the MCSL element as root.

- The Access element contains rules for some or all of the roles in the system that specifies which workspaces each role gives access to.
- The Environment element contains specifications of which environment contributions are present and from which database entities they are obtained.
- The Configuration element contains specifications for the dictionaries and update sites location.

Namespace and Root Tag

Namespace

The namespace of the MCSL format is as follows:

`http://www.deltek.com/ns/mcs1`



In this document, the MCSL namespace is the default namespace to avoid namespace prefixes on tags.

<MCSL>-Tag

The MCSL tag is the root tag of the MCSL specification.

Attribute Name	Type	Usage
Version	Key	This is the MCSL version. This property must be specified.

Access Specification

Workspace Access Model

The access model for the workspace access control such that if at least one of a user's roles gives access to a workspace, the user has access to the workspace. This means that each of the roles that a user has can give access to any number of workspaces, but not restrict the access to workspaces.

Outer <Access>-Tag

The outer Access tag is placed directly in the root MCSL-tag. It serves as a container for the set of role-access specifications in the system. The access tag contains any number of Role tags.

<Role>-Tag

The Role tag specifies the access privileges for a role in the system. The MCSL specification has, at most, one Role tag for each role in the system. The specification cannot contain multiple specifications for a role.

Each role has a compulsory inner Access tag that contains the specification of which workspaces the role may access.

Attribute Name	Type	Usage
Source	Key	This attribute specifies the role to which this Role tag refers. For the Role tag to have an effect, the role must be defined in the system. Roles are defined in Groups in the Maconomy system.
Name	Key	(Optional) This attribute refers to name of this role access specification.

Inner <Access>-Tag

The inner Access tag is placed in a role to specify the workspaces to which the role gives access. The Access tag can contain the list of these workspaces.

Attribute Name	Type	Usage
All	Boolean	(Default: false) This attribute specifies that the role gives access to all workspaces in the system, except those in the Exclude tag of this Access tag.

The following example grants users with role R access to workspaces WS1 and WS2:

```
<Role source="R">
  <Access>
    <Workspace source="WS1" />
    <Workspace source="WS2" />
  </Access>
</Role>
```

```

    </Access>

  </Role>

```

You can also use the all-attribute with the Exclude tag to specify that a role gives access to all workspaces, except for a given list of workspaces. The following example grants users with role R access to all workspaces except WS1 and WS2:

```

<Role source="R">

  <Access all="true">

    <Exclude>

      <Workspace source="WS1" />

      <Workspace source="WS2" />

    </Exclude>

  </Access>

</Role>

```

<AllRoles>-Tag

The AllRoles tag is similar to the Role tag. It is used to describe access rules that span all roles. This tag has the same content as the Role tag.

<Workspace>-Tag

The Workspace tag is used in an Access tag or an Exclude tag to reference a workspace. For the reference to have an effect, the system must contain a workspace with the specified name.

Attribute Name	Type	Usage
Source	Key	This attribute indicates the name of the referenced workspace. The value must correspond to the name attribute of the referenced workspace.
Name	Key	(Optional) This attribute refers to name of this workspace reference.

<Exclude>-Tag

The Exclude tag is an optional tag that is used in the Access tag in a role specification. The Exclude tag contains zero or more Workspace references that represent exceptions from the list of workspaces in the Access rule.

The following example grants users with the role R access to workspace WS1 (the user does not get access to WS2 through the role R):

```

<Role source="R">

  <Access>

    <Exclude>

```

```

        <Workspace source="WS2" />
    </Exclude>

    <Workspace source="WS1" />
    <Workspace source="WS2" />

</Access>

</Role>

```

The Exclude tag is most useful in combination with the all attribute:

```

<Role source="R">

    <Access all="true">

        <Exclude>

            <Workspace source="WS1" />

            <Workspace source="WS2" />

        </Exclude>

    </Access>

</Role>

```

Scope of Exclude

Note that the Exclude tag does not prevent access to a workspace if the user has access to the workspace through another role. In the following example, a user with roles R1 and R2 has access to WS1:

```

<Role source="R1">

    <Access>

        <Workspace source="WS1" />

    </Access>

</Role>

<Role source="R2">

    <Access>

        <Exclude>

            <Workspace source="WS1" />

        </Exclude>

    </Access>

</Role>

```

Environment Specification

<Environment>-Tag

The Environment tag is placed directly in the root MCSL tag. It is designed to specify the contributions that are available in an environment that contains variable values that are used in various functions of the Maconomy system. The Environment tag contains an optional Define tag and any number of Binding tags.

<Binding>-Tag

The Binding tag corresponds to a single database query, whose results are contributed to the current calculated environment entries. To specify the Binding tag correctly, a valid entity name for query and namespace expression must be provided such that there is a correct translation between the query result and environment contribution.

Attribute Name	Type	Usage
entity	Key	This attribute refers to the database entity that contributes entries to the environment.
namespace	Expression	This attribute refers to the environment path expression under which the current contributions are placed.

The namespace is semantically equivalent to the environment path except for the separators, which are a colon (:) in the binding namespace and a dot (.) in the environment path. Expression as a namespace type is used because environment contributions are placed under a path value, which is a result of evaluating an expression.

Each output query row of the system parameters contributions is placed under a different path, which contains, as a last sub-path, the value of an **envVar('InternalName')** function call. This function returns the value of a current processed row's column value, which is set in the context environment.

Thus, it is impossible to specify the path in advance, but it is possible to evaluate it after the query row environment contribution is obtained and appended to the context for the namespace expression evaluation. To support the need to specify the namespace expression easily, a new placeholder expression type has been introduced.

The values specified in this attribute are treated as constant expressions with the special escape sequence **^{}** that treats everything located in the curly braces as an expression for evaluation. For example, the expression "ab^{2+3}c" is evaluated to "ab5c."

Example of Syntax Use

```
"system:parameters:^{envVar('InternalName')}" .
```

<Restriction>-Tag

The Restriction tag is used to provide the condition for filtering rows of the query that correspond to the Binding tag. The condition can be specified as an attribute of the Restriction tag or as a child Condition tag.

Attribute Name	Type	Usage
condition	Expression	(Optional) This attribute refers to the restriction condition that is used for a query.

<Condition>-Tag

The Condition tag is an optional tag that can be used instead of the condition attribute of the Restriction tag. The condition expression is placed directly within the Condition tag root element.

Binding <Fields>-Tag

The Binding Fields tag compresses a set of fields that are translated into a list of selection columns that are used in the query that corresponds to a given Binding tag.

<Field>-Tag

The Field tag corresponds to a single column in the entity. It is used in the Fields tag as a root tag, as a single column that is part of the specified query set. As an alternative, an option has been added to specify the value directly in the specification, allowing contributions that are not obtained from the database. This is done by specifying a pair of attributes; the first one is called **name** and corresponds to the contribution name, and the second one is called **valueString** or **value** depending on the type that is used.

It is only possible to specify either a **source** attribute or a pair of **name** and **valueString** or **value** attributes. Any other combination of attributes is considered an error.

Attribute Name	Type	Usage
source	Key	This attribute refers to the column name of Binding's entity.
name	Key	This attribute refers to the name of the contribution.
valueString	Expression	This attribute refers to the value for a given name of the placeholder string type.
value	Expression	This attribute refers to the value for a given name of the general expression type.

<Define>-Tag

The Define tag is an optional tag that is used to define the named groups of fields that can be referenced, reducing the number of Field tags for Bindings that share a common set. It contains any number of define Fields tags. It serves as a facility that simplifies the process of specification writing.

Define <Fields>-Tag

The Define Fields tag is placed directly in the Define tag as a root tag and used to map a unique name to a set of fields that can be referenced from the Reference Fields tag.

Attribute Name	Type	Usage
name	Key	This attribute refers to the name of this Fields group for reference.

Reference <Fields>-Tag

The Reference Fields tag can only be used as a child of the Fields tag to insert all fields from the corresponding Define Fields tag. It can be a child of either the Define Fields tag or Finding Fields tag.

Attribute Name	Type	Usage
ref	Key	This attribute refers to the reference to the name of the Define Fields tag group.

Example of Environment Specification

```

<Environment>
  <Define>
    <Fields name="MyFields1">
      <Field name="a0" valueString="v0" />
      <Field source="a1" />
      <Fields ref="MyFields2" />
    </Fields>
    <Fields name="MyFields2">
      <Field source="b1" />
      <Fields ref="MyFields3" />
      <Field source="b2" />
    </Fields>
    <Fields name="MyFields3">
      <Field source="c1" />
      <Field source="c2" />
      <Field source="c3" />
    </Fields>
  </Define>

```

```
<Binding entity="A" namespace="a">
  <Fields>
    <Fields ref="MyFields1" />
  </Fields>
</Binding>
<Binding entity="B" namespace="b">
  <Fields>
    <Fields ref="MyFields2" />
  </Fields>
</Binding>
<Binding entity="C" namespace="c">
  <Fields>
    <Fields ref="MyFields3" />
  </Fields>
</Binding>
</Environment>
```

This specification results from the following query to environment path translation:

- Query on entity A with column selection a1,b1,b2,c1,c2,c3 inserted into environment "a" path
- Query on entity B with column selection b1,b2,c1,c2,c3 inserted into environment "b" path
- Query on entity C with column selection c1,c2,c3 inserted into environment "c" path



Environment path "a" is additionally contributed with a contribution to the sub-path "a0" having the value "v0" from the specification.

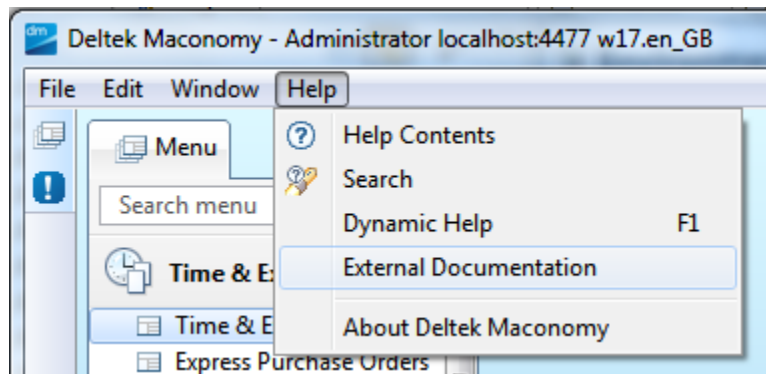
Environment – Reserved Bindings

The environment area of MCSL is used to specify generic bindings that clients can access. Certain bindings are, however, reserved for particular clients. The Workspace Client has a special built-in interpretation of the reserved bindings that described in this section. Other clients, such as Navigator, may impose special semantics on other bindings at a later stage.

External Documentation Link

The primary means of providing end-user documentation in the Workspace Client is through the Help panel (accessed by pressing F1). Adding content to this section is, however, a fairly complex process and requires detailed technical understanding of the Workspace Client architecture.

A simpler and more discrete solution is to provide an external documentation link to an external web site, such as a company intranet. The link appears as a clickable item in the Help group in the menu bar (shown in the following figure).



The preceding link appears if the following reserved binding is added to the MCSL environment specification. The value of the **Url** field determines the external documentation link's target. This can be any URL that is accessible from a standard browser on the end user's machine.

```
<Binding namespace="documentation">

  <Fields>

    <Field name="Url" valueString="www.kona.com" />

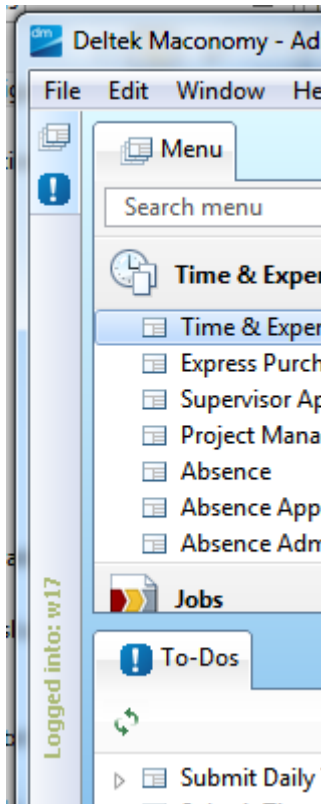
  </Fields>

</Binding>
```

Menu Dock Decorations

A typical use case during implementation projects is to switch between different systems with the same Workspace Client. To help a user to distinguish between the different systems, the vertical bar left of the menu, known as the *Menu Dock*, can be decorated with data-dependent text. This can, for instance, be used to show the system shortname such that a user can distinguish between a Test and a Production system.

The following figure and code sample show an example configuration. Notice that the Expression Language can be used to make the text data-dependent using the normal placeholder syntax also known from MDML and MWSL.



```
<Binding namespace="client:menudock">
  <Fields>
    <Field name="Title" valueString="^{ 'Logged into: ' +
shortname() }"/>
    <Field name="Color" valueString="rgb=#abba76" />
    <Field name="TopToBottom" valueString="false" />
    <Field name="FontHeight" valueString="8" />
  </Fields>
</Binding>
```

You can specify another color by assigning another field called "Color" to a valueString that has one of the two formats:

```
rgb=num,num,num (where num is a decimal number in the
range 0-255)
```

OR

`rgb=#xxyyzz` (where `xx`, `yy` and `zz` are two-digit hexadecimal numbers in the range 00-FF).

Other attributes that you can specify are:

- `FontHeight`, which indicates the font height to use (default is 12).
- `TopToBottom`, which indicates whether the text is drawn top-to-bottom (this is the default) or bottom-to-top (if the value is specified as false).

Configuration Specification

The configuration area of MCSL is used to specify the dictionaries and update sites specifications.

<Configuration>-Tag

The Configuration tag is used in the root MCSL-tag to encapsulate the configuration specification.

Dictionaries Specification

You can specify the list of dictionaries that will be enabled to the client to use (providing the localization into the dictionaries' languages). Specification of the dictionaries is made in MCSL because the dictionaries are contained in the Coupling Service installation.

First, there is a shorthand notation to enable all available dictionaries.

```
<Dictionaries all="true" />
```

Second, you can restrict the set of available dictionaries, so that the client can use a defined subset.

```
<Dictionaries>
  <Dictionary source="id1" />
  <Dictionary source="id2" />
  ...
  <Dictionary source="idn" />
</Dictionaries>
```



The **id1**, **id2**..., **idn** variables refer to the identifiers of the dictionaries that correspond to the names of the dictionaries files.

Update Sites Specification

The update sites specification is used to define the locations of update sites from which the latest updates for the client can be downloaded.

<UpdateSite>-Tag

An instance of the UpdateSite tag is used to define a single update site location. It has a single attribute **location** that is used for this purpose.

Attribute Name	Type	Usage
location	Key	This attribute refers to the URL that specifies the update site location.

Example of Configuration Specification

```
<Configuration>

  <Dictionaries all="true"/>

  <UpdateSites>

    <UpdateSite location="http://<some location>:1234/" />

    <UpdateSite      location="http://<other location>:1234/"
  />

    ...

  </UpdateSites>

</Configuration>
```

Pop-Ups Specification

Pop-ups specification is used to define the filtering and sort order of the pop-up values. The specification consists of a **Popups** element that specifies global settings for pop-ups filtering and sorting that consists of any number of **Popup** elements that override the global specification for a particular pop-up type.

<Popups>-Tag

An instance of the **Popups** tag is used to define the pop-up value filtering and sort order for all of the pop-ups. It has the following attributes.

Attribute Name	Type	Usage
restrain	Expression	This attribute refers to the Boolean expression written in the Expression Language that filters out all of the pop-up values that satisfy it. It applies to all of the pop-up types.
order	Order	<p>This attribute refers to ordering that will be applied to the pop-up values for all of the pop-up types. Valid values are:</p> <ul style="list-style-type: none"> Ascending — Alphabetical sort order in increasing order. Descending — Alphabetical sort order in decreasing order. Inherent — No sorting applied, the order is as passed by the server. Standard — One of the preceding values as specified in the configuration; the default is Inherent.

<Popup>-Tag

An instance of the Popup tag is used to define the pop-up value filtering and sort order for a specific pop-up type that is specified by the source attribute. The restrain and order attributes override the values in the Pops element if they have been specified.

Attribute Name	Type	Usage
source	Key	This attribute refers to the pop-up type name.
restrain	Expression	The same as for the restrain attribute in the Popup element, except that it only applies to the pop-up that has the type that is specified in the source attribute.
order	Order	The same as for the order attribute in the Popup element, except that it only applies to the pop-up that has the type that is specified in the source attribute.

Example of Configuration Specification

```
<Configuration>

  <Dictionaries all="true"/>

  <UpdateSites/>

  <Pops restrain="startsWith(popup.literal,'A') "
order="inherent">


    <Popup source="CountryType "
restrain="startsWith(popup.literal,'S') or
startsWith(popup.literal,'B') " order="descending" />

    <Popup source="PostingType "
restrain="startsWith(popup.literal,'B') " order="descending" />

    <Popup source="ZipCodeType "
restrain="startsWith(popup.literal,'0') " />

  </Pops>

</Configuration>
```



Deltek is the leading global provider of enterprise software and information solutions for professional services firms, government contractors, and government agencies. For decades, we have delivered actionable insight that empowers our customers to unlock their business potential. Over 14,000 organizations and 1.8 million users in approximately 80 countries around the world rely on Deltek to research and identify opportunities, win new business, optimize resource, streamline operations, and deliver more profitable projects. Deltek – Know more. Do more.®

deltek.com