Deltek Maconomy BPM Extension Guide

Business Performance Management

— *covering version 2.6.4*

Copenhagen, 2024

Edited by

Asger Eir

©2013–2024, *Deltek Inc.*

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 2024.

© 2013-2024 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.

©Deltek Inc. 2013–2024, All Rights Reserved

Contents

1 Introduction **1**
- 1.1 Types of extensions . . . 1
- 1.2 Upgradabiity . . . 2

2 WebI Report Extensions **3**
- 2.1 Steps for extending a report . . . 3
- 2.2 Considerations when upgrading . . . 4

3 Universe Extensions **5**
- 3.1 Prerequisites . . . 5
- 3.2 Steps for extending a universe by the *Link Method* . . . 6
- 3.3 Steps for extending a universe by the *Copy Method* . . . 6
- 3.4 Naming conventions . . . 7
- 3.5 Considerations when upgrading . . . 7

4 ETL Extensions **9**
- 4.1 ETL copying and reorganization . . . 9
- 4.2 Extensions when upgrading . . . 10
- 4.3 Naming conventions . . . 10
  - 4.3.1 General rules . . . 11
- 4.4 Extension cases . . . 12
  - 4.4.1 Adding an extension fact table . . . 12
  - 4.4.2 Adding an extension dimension table for an extension fact table . . . 12
  - 4.4.3 Adding an extension dimension table for a standard fact table . . . 12
  - 4.4.4 Adding an extension dimension field to a standard fact table . . . 12
  - 4.4.5 Adding an extension fact field to a standard fact table . . . 13

©Deltek Inc. 2013–2024, All Rights Reserved

Chapter 1

Introduction

This document is a guide in how to do BPM extensions. An extension is a customized part of the BPM product that is developed to fit customer's needs. The BPM product basically consists of *WebIntelligence Report Documents*, *Universes*, *Pentaho ETL jobs and transformations*, and *Data Exports using the Java Extensions Framework*. Changes to workspaces that provide selection criteria for reports embedded in the Workspace Client, can also be considered extensions but these are not covered by this document.

1.1 Types of extensions

By an extension, we shall understand an addition to the standard BPM product delivered with the RPU (*Reporting Packing Unit* and an APU *Application Packing Unit*. An RPU contains BIAR files containing standard BusinessObjects universes and standard WebIntelligence report documents. Furthermore, the RPU contains standard Pentaho PDI ETL for the BPM Analysis data warehouse. A BPM extension can either be developed using these standard parts as foundation, or they can be development of whole new universes, reports, etc. that are not using the standard ones as foundation. They can also be a mixture. E.g. an extension report can be developed without using a standard report as foundation but still reference standard universes or extensions to standard universes. The Maconomy APU contains standard workspaces that embed BPM report documents, as well as standard extesions that offer data export functionality. For the latter, we refer to the *BPM Data Export Extension Guide*.

The different types of extensions are:

A report extension basically means creating a copy of a standard report document and modifying it. It can also be creating a whole new report document in WebIntellingence. The report can use standard universes or universe extensions.

A universe extension is a universe which uses a standard universe as foundation for providing an enhanced universe with more objects, filters, tables, etc. It can also

be creating a whole new universe.

An ETL extension is basically modifying the standard ETL to include additional ETL jobs and transformations.

1.2 Upgradabiity

When doing extensions, one of the important aspects to keep in mind is whether it is easy to keep the changes when upgrading Maconomy and BPM. We shall distinguish between extensions that are *Upgradable* from those that are *Non-Upgradable*.

An extension is *Upgradable* if it does not require reimplementation (or close to no corrections) in order to—after the upgrade—be fully functional and capable to benefit from the upgraded BPM packages installed.

The following table, shows which sorts of extensions (and combinations of extensions) that are upgradable and which that are not:

Extension	Upgradable
Report extension. Standard universe	Upgradable†
Report extension. Universe extension (Link Method)	Upgradable††
Report extension. Universe extension (Copy Method)	Non-Upgradable† † †
Universe extension. Link Method	Upgradable
Universe extension. Copy Method	Non-Upgradable
ETL extension	Non-Upgradable

†) A report extension on a Standard universe is considered upgradable because the report does not have to be reimplemented and it will benefit from all corrections made to the upgraded universes that it applies. Furthermore, enhancements done to the upgraded universe are possible to utilize by doing simple changes to the report. †† A report extension on a universe extension is likewise upgradable if the universe extension was made using the Link Method; see Section 3.2. The reason is that any correction done to the upgraded universes that the report applies, will directly be part of the extension report. † † † A report extension on a universe extension is, however, considered non-upgradable, if the universe extension was made using the Copy Method. The reason is that corrections done to the upgraded universes that the report's applied universes originally were copies of, will not apply in the report.

©Deltek Inc. 2013–2024, All Rights Reserved

Chapter 2

WebI Report Extensions

Report extensions can be done in two different ways:

- By creating a new WebIntelligence report document.
- By modifying a standard WebIntellingence BPM report document.

Such report documents, we shall refer to as *extension reports*.

R All extension reports must be stored in a CMS folder different from the standard folder and any sub-folder to the standard folder: `/Business Performance Management`.

2.1 Steps for extending a report

To create an extention of a standard BusinessObjects BPM report, follow these step:

1. Open a browser and log into the SAP BusinessObjects BI LaunchPad
2. In the BI LaunchPad, navigate to the folder containing the standard report document to be extended.
3. Open the report document.
4. Save the report document to a suitable extension folder. **Note:** do not save extension reports in the standard folder structure as it creates confusion on which reports in the CMS that are part of the standard delivery.
5. Apply the desired changes and save the report document again.

R Some report changes can only be done from WebIntelligence. An example is adding additional result objects to a query or adding new queries. To do such changes, right-click the extension report and choose `modify...`. Alternatively, open the extension report from WebIntelligence.

2.2 Considerations when upgrading

When upgrading to a new BPM package, it is always a possibility that the universes in the new release have been cleaned for deprecated objects that are being used in extension reports. It is also a possibility that incorrect behaviour in objects has been assumed in a customisation and that fixes of such bugs makes the extention report give unforeseen results.

When upgrading, consider the following:

- is the report using objects that have earlier been deprecated? For each release, the RPU lists the object descriptions of each object in a universe in HTML files; one for each universe[1]. If an object is deprecated, it is stated in the object descriptions and the object title is striked through.
- consult release notes concerning fixed bugs.
- consult release notes concerning changes in universe functionality.
- if the report uses many objects that were deprecated in previous releases and thus may be removed in the new release, consider making a copy of the universe and gradually resolve the issues. Typically, objects are only removed if they use database fields that are no longer in use, or if the functionality in the universe has been restructured. In the latter cases, other objects should be available for providing the same reporting capabilities.

[1]See more about the content of RPUs in the BPM Administrative Guide.

©Deltek Inc. 2013–2024, All Rights Reserved

Chapter 3

Universe Extensions

Universe extensions can be done in two different ways:

- By creating a new universe which links to a standard universe. We shall call this the ***Link Method***. Use this method if you which to add objects or other functionality to an already existing standard universe. This type of universe extension is recommended because it is upgradable.
- By creating a new universe as a copy of a standard universe. We shall call this the ***Copy Method***. Only use this type of universe extension if it is not possible or convenient to use the one above. The reason is that making a universe copy does not provide a universe extension which is upgradable. This means more maintainance of the universe.

Such universes, we shall refer to as *extension universes*.

All extension universes must be published to a CMS folder different from the standard folder and any sub-folder to the standard folder: `Universes/Business Performance Management`. The suggested folder for extension universes is: `Universes/Extensions/`

3.1 Prerequisites

The prerequisites for creating a universe extension are:

- A BPM installation has been made such that the CMS contains the standard universes.
- The connection has been set at least for the universe to extend.
- The standard universes have been retrieved to universe files using SAP BusinessObjects Information Design Tool.

3.2 Steps for extending a universe by the *Link Method*

To create an extention of a universe, follow these step:

1. Open the SAP BusinessObjects Information Design Tool.
2. Optionally, create a new local project by clicking File»New...» Project.
3. In the Repository Resources view, right-click the standard universe to base the extension on and select Create Linked Universe.
4. Select the local project. Click Next.
5. Accept using the existing connection. Click Next.
6. Enter a name for the extension universe.
7. Enter a name for the Data Foundation (recommended to be the same). Click Finish.
8. Publish the universe

You are now ready to add your own tables, folders, objects or other universe items. All items originating from the core Standard universe, will appear dimmed and cannot be edited.

3.3 Steps for extending a universe by the *Copy Method*

To create an extension of a universe, follow these steps:

1. Open the SAP BusinessObjects Information Design Tool.
2. Optionally, create a new local project by clicking File»New...» Project.
3. In the Repository Resources view, right-click the standard universe to base the extension on and select Retrieve Universe.
4. Select the local project.
5. Optionally, enable Save for all sers (no security). This setting is recommended if the universe files are to be shared on different servers. Click Finish
6. Optionally, move the retrieved universe and connection files from the timestamp folder created, to the desired location in the local project.
7. Optionally, change the name of the universe.
8. Publish the universe. **Beware:** It is important that the CMS folder to which the universe is published, is another one than the standard universe it was based on. Else, the extension universe could name clash when importing a BIAR containing the standard universes.

©Deltek Inc. 2013–2024, All Rights Reserved

3.4 Naming conventions

All extension universes should be prefixed “c_”.

3.5 Considerations when upgrading

When upgrading BPM, a new set of universes are installed in the CMS. Whether corrections are needed and how many, depends on the approach that has been used.

Link Method: Universe extensions using the Link Method, will automatically be upgraded without any need for corrections. It is, though recommended to keep the extension part of the universe updated so they stay aligned with the database schema currently installed. Use Integrity Check and Refresh Structure to ensure consistency to the database schema.

Copy Method Universe extensions using the Copy Method, will *not* be upgraded. They will remain as they are and thus be behind in version. They must be maintained and kept up-to-date manually. It is recommended to update such extension universes regularly so they stay aligned with the database schema currently installed. Use Integrity Check and Refresh Structure to ensure consistency to the database schema.

In main versions, tables, objects and other universe items may have been removed. If items in the extension universe are referencing removed items, the extension universe will not be valid. For universe extensions using the Link Method, the link should be removed prior to the upgrade and re-established after upgrading. Some items may need to be corrected if they reference fields that have been removed, or similar. For universe extensions using the Copy Method, a more thorough check and test of the universe should be applied.

For both approaches, the following list are considerations to be aware of:

- Fields and tables in Maconomy, may have been removed or renamed. This will make the extension universe invalid. Apply Refresh Structure to update the universes meta information about the database schema and run Integrity Check to make sure of consistency.
- In rare cases, fields may change semantics. This may change the behaviour of universes and reports. Read release notes to make sure no changes have been made that could make the extension universe invalid.

Since universe extensions using the Link Method are typically having a much small extension contribution (the rest is provided by the standard universe), there is much less chance that the above sorts of chances are needed. The standard universes can be considered updated to comply with any changes done to the database schema or semantics of database fields.

©Deltek Inc. 2013–2024, All Rights Reserved

©Deltek Inc. 2013–2024, All Rights Reserved

Chapter 4

ETL Extensions

4.1 ETL copying and reorganization

Extensions to tables and functionality in BPM Analysis, is done by making changes to the ETL transforming data, and the jobs `CreateTablesAndViews` and `DropTablesAndViews` for building or dropping the data warehouse. There are two strategies for doing this:

- *ETL extension by calling* in which the standard job `LoadData` is replaced by an extension version.
- *ETL extension by sequencing* in which the standard job `LoadData` and an extension version `c_LoadData` are set to be executed in sequence.

ETL extension by calling is beneficial if the extension introduces new fields and tables that are independent of the data loaded into the standard tables by `LoadData`. It is the simplest approach and does not require loading more than once, which can have an impact on performance. Also, it is more flexible in the sense that parts of the standard delivery can (if really necessary) be removed from `c_LoadData` if not desired. However, it is likely that more extensions need to be re-established if the ETL structure of the standard delivery changes. ETL extension by sequencing is beneficial if the extension needs to utilize data loaded into the standard tables by `LoadData`. An example is the introduction of aggregated data in dedicated extension tables. An *extension table* is a table in the Data Warehouse, that is introduced in relation to an ETL extension. The extension approach is not that sensitive against changed structure in the standard delivery, though some changes may need to be done; e.g. if a table is taken out of the delivery and the extension uses this table. However, the approach is not that flexible and `LoadData` will always run first so `c_LoadData` should not include things already done in `LoadData`. With the former approach, `c_LoadData` is the job to be called when scheduling loads. `LoadData` is thereby replaced and not called. With the latter approach, `LoadData` and `c_LoadData` are to be called in sequence. [1]

[1] Note, that `LoadData` finalizes by handling e.g. the ETL log. Running a `c_LoadData` after that does

Similar must be done with `CreateTablesAndViews` and `DropTablesAndViews`.

R The standard job `LoadData` calls transformations that maintain data in `DWH_MANAGEMENT` including information about date ranges for incremental loads. If doing extension by sequencing, transformations should reuse the date (current_max_date) in `DWH_MANAGEMENT` in its restriction for incrementally loaded data.

4.2 Extensions when upgrading

When upgrading, a number of steps need to be followed in order to make the re-introduction of the extensions as smooth as possible. The Pentaho™ repository is used as a central storage in which the new standard delivery and the current extensions are merged. The overall steps are:

- Export the extension ETL folder
- Remove the Pentaho™ repository
- Create a new Pentaho™ repository
- Import the standard ETL delivery
- Import the extension ETL

There can be several variations of the above.

4.3 Naming conventions

In order to easily distinguish standard deliveries from extensions, the following naming conventions should be observed:

Extension tables Prefix table names with "c_" or "C_".
Example: `C_F_Job_Entry`

Extension indexes Prefix indexes with "c_" or "C_".
Example: `C_D_JOB01`

Extension ETL Prefix job and transformation names with "c_" or "C_".
Place all extension jobs and transformation in a dedicated folder named "Extension" or "extension".[2]

In the following some general procedures and rules for doing extensions to the ETL of the data warehouse, are described. The rules apply only to ETL which is part of BPM Analysis.

not include the jobs run in the ETL log. With the current organization of the standard ETL, minor changes may be needed, depending on the requirements of the customer.

[2] the previously suggested folder name "Custom" is also acceptable.

©Deltek Inc. 2013–2024, All Rights Reserved

4.3.1 General rules

- All extensions to ETL (i.e. jobs or transformations) should be done by extension jobs and transformations. An *extension job* or a *extension transformation* is a job or transformation that is introduced as part of an ETL extension.
- These extension jobs and transformations must be placed in a folder called "extension" or "Extension" in the repository.
- No standard jobs or transformations should be changed.
- The folder structure (besides that in the extension folder) should remain unchanged.
- New jobs/transformations for loading extension-defined data should be scheduled or manually run after the standard `LoadData` or an extension version of `LoadData` can be made.
- When upgrading, extensions may need to be reconsidered:
  - An upgrade may introduce standard functionality which previously has been implemented by extensions. It is recommended to include the standard functionality in favor of the extensions, if they fully satisfy the needs of the customer.
  - An upgrade may change field names or semantics of fields. Consequently, extensions need to be adjusted carefully.
  - An upgrade may declare fields or tables as deprecated (e.g. if they are replaced by other or not included in Maconomy anymore). Extensions relying on such fields and tables should be considered changed. In some situations it may be convenient for the customer to change the extensions such that they still rely on old functionality; e.g. deprecated fields. Deprecated fields or tables that are removed by the upgrading process can then be re-introduced in extension tables. Note that a dump of the content of the deprecated table then needs to be done as part of the upgrading process.
  - Carefully document the steps done in the extension such that these can easily be reconsidered and possibly re-done after an upgrade. Primarily, document the following:
    * Which standard tables the extension tables are copies of.
    * Which fields have been added to the copies of the standard tables.
    * The links that have been made.

©Deltek Inc. 2013–2024, All Rights Reserved

4.4 Extension cases

4.4.1 Adding an extension fact table

1. Create a new extension fact table.
2. Add links to dimensions (standard and/or extension).
3. Create a new extension ETL job (or modify an existing extension ETL job) for handling loads into the extension table.

4.4.2 Adding an extension dimension table for an extension fact table

1. Create a new extension dimension table.
2. Add links from the extension fact table to the extension dimension table.
3. Create a new extension ETL job (or modify an existing extension ETL job) for handling loads into the table.

4.4.3 Adding an extension dimension table for a standard fact table

1. Create a new extension dimension table.
2. Create a copy of the standard fact table. This is now an extension fact table
3. Re-establish links from the extension fact table to the standard dimension tables
4. Add a link from the extension dimension table to the extension dimension table.
5. Create a new extension ETL job (or modify an existing extension ETL job) for handling loads into the extension tables.

4.4.4 Adding an extension dimension field to a standard fact table

1. Create a copy of the standard dimension table. This is now an extension dimension table.
2. Extend the extension dimension table with the field desired.
3. Create a copy of the standard fact tables for which the new dimension field is to be traced. These are now an extension fact tables.
4. Re-establish links from the extension fact tables to the standard dimension tables.
5. Add a link from the extension fact tables to the extension dimension table.
6. Create a new extension ETL job (or modify an existing extension ETL job) for handling loads into the extension tables.

©Deltek Inc. 2013–2024, All Rights Reserved

4.4.5 Adding an extension fact field to a standard fact table

1. Create a copy of the standard fact table. This is now an extension dimension table.
2. Extend the extension fact table with the field desired.
3. Re-establish links from the extension fact table to the dimension tables.
4. Create a new extension ETL job (or modify an existing extension ETL job) for handling loads into the extension tables.

©Deltek Inc. 2013–2024, All Rights Reserved

©Deltek Inc. 2013–2024, All Rights Reserved