IBM Cognos Controller Financial Analytics Publisher

Version 10.1.0

User Guide

Product Information

This document applies to IBM Cognos Financial Analytics Publisher version 10.1.0 and may also apply to subsequent releases. To check for newer versions of this document, visit the IBM Cognos Information Centers (http://publib.boulder.ibm.com/infocenter/cogic/v1r0m0/index.jsp).

Copyright

Table of Contents

Part 1: Appendices

Introduction

This document is intended for use with IBM® Cognos® Controller Financial Analytics Publisher.

Cognos Controller Financial Analytics Publisher allows you to connect the IBM® Cognos® Controller database to the Cognos Controller Financial Analytics Publisher client so that you can publish data in a TM1® cube. The solution is optimized for many simultaneous users, and suits both legal and management reporting.

For installation instructions, see the *IBM Cognos Controller Installation and Configuration Guide*.

Audience

This document will help Controller application administrators and the report writers to use Cognos Controller Financial Analytics Publisher.

Finding information

To find IBM® Cognos® product documentation on the web, including all translated documentation, access one of the IBM Cognos Information Centers at http://publib.boulder.ibm.com/infocenter/cogic/v1r0m0/index.jsp. Updates to Release Notes are published directly to Information Centers.

You can also read PDF versions of the product release notes and installation guides directly from IBM Cognos product disks.

Using quick tours

Quick tours are short online tutorials that illustrate key features in IBM Cognos product components. To view a quick tour, start IBM Cognos Connection and click the **Quick Tour** link in the lower-right corner of the Welcome page. Quick Tours are also available in IBM Cognos Information Centers.

Forward-looking statements

This documentation describes the current functionality of the product. References to items that are not currently available may be included. No implication of any future availability should be inferred. Any such references are not a commitment, promise, or legal obligation to deliver any material, code, or functionality. The development, release, and timing of features or functionality remain at the sole discretion of IBM.

Accessibility Features

This product does not currently support accessibility features that help users who have a physical disability, such as restricted mobility or limited vision, to use this product.

Chapter 1: Getting started

To get started using IBM® Cognos® Controller Financial Analytics Publisher, you need to:

❑ Select database type

❑ Define service settings

❑ Define source

❑ Add data marts

❑ Filter log information

Selecting a database type

When you log on to the IBM® Cognos® Controller Financial Analytics Publisher client for the first time, you need to enter the database type, which can be either Microsoft® SQL Server, Oracle or DB2®.

Steps

1. Click **Start > IBM Cognos Controller Financial Analytics Publisher**.
2. In the **FAP Connect** window, provide the following information:

   For MS SQL Server:

   - Database type - select which type of database server you want to use
   - Server - select the database server name
   - Database - select the name of the Financial Analytics Publisher database
   - Username - enter a valid Financial Analytics Publisher user
   - Password - enter the valid Financial Analytics Publisher user password

   For Oracle:

   - Database type - select which type of database server you want to use
   - Data source - service name
   - Username - enter a valid Financial Analytics Publisher user
   - Password - enter the Financial Analytics Publisher user password

   For IBM DB2:

   - Database type - select which type of database server you want to use
   - Server - select the database server name

- Database - select the name of the Financial Analytics Publisher database
- Username - enter a valid Financial Analytics Publisher user
- Password - enter the valid Financial Analytics Publisher user password

The next time you log on, you only need to enter the password.

3. Click **Log On**.

Defining service settings

Service settings allow you to manage factors that might have an impact on performance and disk space. You can modify how often the different service settings should be run by using the **Service Settings** tab.

Steps

1. In the **IBM Cognos Controller Financial Analytics Publisher** window, click the **Service Settings** tab.
2. In the **Clock Interval** list, select how long the interval should be between each time the service is run.
3. In the **Log file items for** list, select how often the log table is purged in the FAP database.
4. In the **Trickle tables purge every** list, select how often the trickle table is purged.

Defining sources

You can define the source, which is used to connect the Cognos Controller database with the data mart. It includes information about the source status, purged date, last purged index, and the Cognos Controller database last moved index.

Steps

1. In the **IBM Cognos Controller Financial Analytics Publisher** window, click the **Sources** tab.
2. Click **New**.
3. In the **Name** field, enter the name of the source.
4. From the **Log level** list, select **Low**, **Normal**, or **High**. **Normal** is selected by default.
5. Select an interval.
6. Optional: in the **Description** field, enter a description of the source.
7. Set the database options:
   - For MS SQL Server, select the database type, the server, and the database you want to use.
   - For Oracle, select the database type, the server, the SID, and the port.

- For DB2, select the database type, the server, and the database you want to use.

8. Enter a valid username and password.
9. To test the new connection, click **Test Connection**.
10. Click **Save**.

    You can change the source settings by clicking the **Edit** button.

Adding a data mart

You can add data marts to define which data you want to publish, and save the information as a definition. You can choose to publish the company structure, start period and actuality. You can edit the definition and add a new company structure when needed. You can also specify which forms are to be published, edit naming conventions for specific structures and select a company details or a consolidated groups cube.

Note: IBM Cognos Financial Analytics Publisher must be used in conjunction with Cognos Controller databases that have verified structures and validated data. Before publishing, verify the structures and validate the data using IBM Cognos Controller.

Steps

1. In the **IBM Cognos Controller Financial Analytics Publisher** window, click the **Data Marts** tab.
2. Click **New**.
3. Set the data mart options:
    - **Name** - enter the name of the data mart
    - **Source** - select a defined source
    - **Log level** - specify the log information level for this particular data mart. It is usually best to use log level **High**.
    - **Interval** - specify how many service clock intervals will elapse until the tasks for this data mart are performed
    - **Specify forms** - specify which forms you want to publish
    - **Naming conventions** - specify which names you want to use
    - **Company details** - publish the company details cube
    - **Consolidated groups** - publish consolidated groups
    - **Description** - enter text that describes the data mart
    - **Company Structure** - select one or more consolidation types for the company structure in the data mart, and enter the period (YYMM) from which the company structure in the data mart will be set up
    - **Start period/Actuality** - decide from when data will be visible in the data mart.

- **Admin host** - specify the name of the server that hosts the TM1 servers
- **Server** - specify the name of a specific TM1 server.
- **Client** - use TM1 Admin
- **Password** - specify a valid password for the TM1 user.

4. Click **Save.**
5. Confirm that the new data mart is visible in the list on the **Data Marts** tab.

   You can change the source settings by clicking the **Edit** button.
6. Select the data mart that you want to run, and click **Start**. The status changes from **Under definition** to **Ready for publish.**

The next time that the FAP Service runs, it detects the **Ready for publish** status, creates the cube, and starts the initial publishing of data. When the initial publishing is finished, the data mart status changes to **Running**. The cube is then updated on a regular basis according to the **Clock Interval** setting on the **Service Settings** tab.

Viewing log information

You can view log information about the status of the data mart and the source. This allows you to keep track of every task that is performed.

Steps

1. Click the **Logs** tab.
2. Under **Filter**, select the filter options:
   - **Type** - you can specify log information type: **Any**, **Log**, **Source** and **Data Mart**.
   - **Object** - you can specify which objects you want to see the log information about.
   - **Severity** - you can specify severity level: **Any Severity**, **Info**, **Error** and **Critical Error**.
   - **Page size** - you can specify a page size of **50**, **100**, **500** or **1000**.
3. On the **View** menu, click **Refresh** to update the log.
4. If you want to update the log information automatically, select the **Update automatically** check box.

Verifying Cognos Controller Financial Analytics Publisher data

Use IBM Cognos Business Intelligence studios, or other TM1 compatible reporting tools to verify the data and the structures in the TM1 cube.

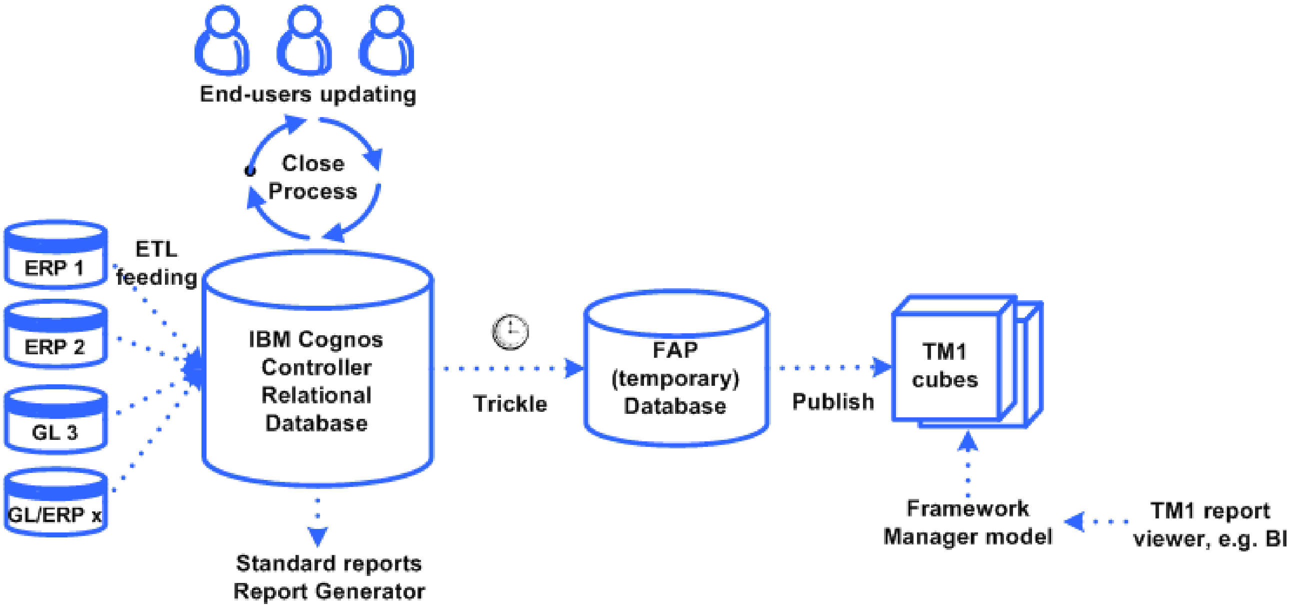
Chapter 2: How Cognos Controller data is transformed into a reporting format

IBM® Cognos® Controller Financial Analytics Publisher (FAP) automates the process of extracting data from IBM Cognos Controller inserting it into IBM Cognos TM1®. From the TM1 cube(s), the Cognos Controller data can be accessed by IBM Cognos Business Intelligence studios, and other TM1 compatible reporting tools.

FAP overview

FAP provides additional ways of solving both in-process reporting requirements, as well as management reporting agendas, using data without any time lags. FAP is delivered with IBM Cognos Controller.

Users want access to consolidated data to perform analytic-style reporting and distribute reports on a global scale. FAP extracts the information from the Cognos Controller relational database and inserts it into a TM1 cube environment. Any changes made to the data in the Cognos Controller database, including those generated by the users or fed from source systems, are automatically propagated through to TM1.



FAP combines two different technologies: a the relational database technology optimized for transactions and data entry and an OLAP technology optimized for reporting and analytics.

The peak load on FAP occurs during the initial publish, during the closing period, when the majority of changes are processed in Cognos Controller, and during the reporting peaks in the cube.

FAP components

FAP consists of four main components:

- FAP client - This is an administrative console where the basic settings are applied, some events are triggered, and the FAP log information can be monitored.
- FAP database - This is a temporary database to which changed facts and data and dimension updates are moved prior to their transformation into the cube. The information in this database is purged at regular intervals.
- FAP service - This moves and transforms the relational Cognos Controller data and the dimensions into the cube.
- TM1 (FAP) cubes - These are the read-only TM1 cubes that contain the Cognos Controller dimensions and trickle-published data.

FAP requires additional server resources for TM1 and the FAP database. For more information about network set up and resource requirements, see the IBM Cognos Proven Practices document *Architecture and Server Sizing for IBM Cognos Controller 8.5* (http://public.dhe.ibm.com/software/dw/dm/cognos/performance/cognos_specific/architecture_and_server_sizing_for_cognos_controller85.pdf).

FAP concepts

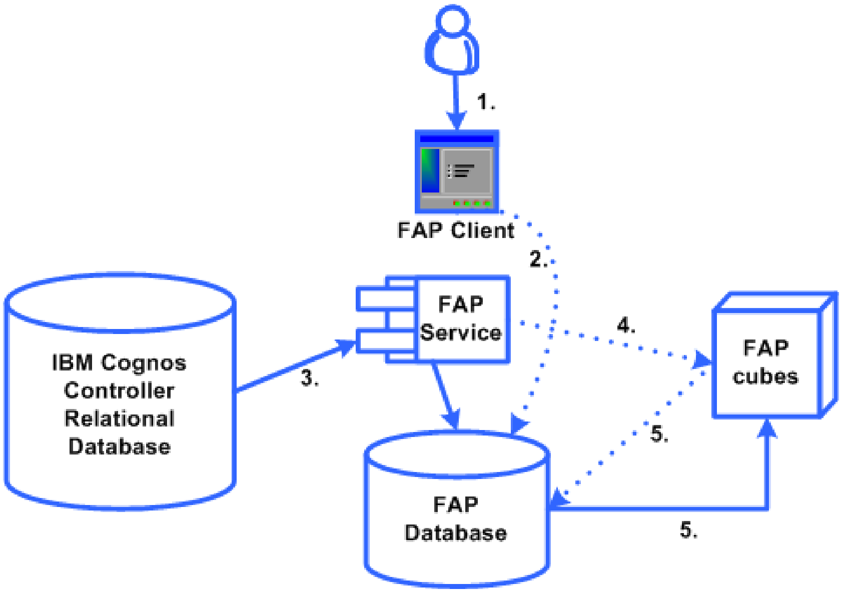
It is important to understand the following process when designing a reporting solution:

- Initial publish
- Fact (data) trickling
- Structure update
- FAP database purging

You can add extension points both before and after each of these processes. For more information, see "FAP extension points" (p. 39).

Initial publish

The initial publish process creates the FAP dimensions and cubes and fills the cubes with facts and data. If the dimensions are already present on the TM1 server, they are emptied and not re-created.



The initial publish process:

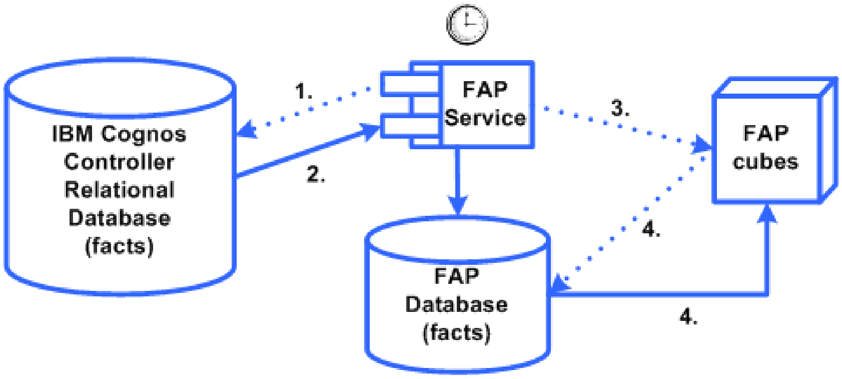
1. Select the Cognos Controller data to publish in the FAP client.
2. Start the initial publish from the FAP client. You can initiate a full or partial initial publish process.
3. The FAP Service transfers facts and dimensions from Cognos Controller to the FAP database.
4. A TM1 API call is initiated, building dimensions and creating the cubes.
5. A TM1 Turbo Integrator (TI) script is run to collect facts and data from the FAP database and fill the cube.

It is recommended that you verify all Cognos Controller structures prior to starting the Initial Publish.

Once the initial publish is completed it is shown in the FAP client log. Please note that a full initial publish may take some time, depending on the volumes to be moved from the source to the target.

Fact trickling

Fact (data) trickling is the repetitive process in which the FAP Service looks for changes (inserts, updates, deletions) in the Cognos Controller database, and updates the FAP cubes accordingly.



The fact trickling process:

1. Based on the clock interval defined in the FAP client, the FAP Service searches for changes in the Cognos Controller database.

2. The FAP Service transfers all changes from the Cognos Controller database to the FAP database.

3. A TI process fetches facts from the FAP database and updates the nodes in the FAP cubes.

Each data trickle cycle generates a log in the FAP client log.

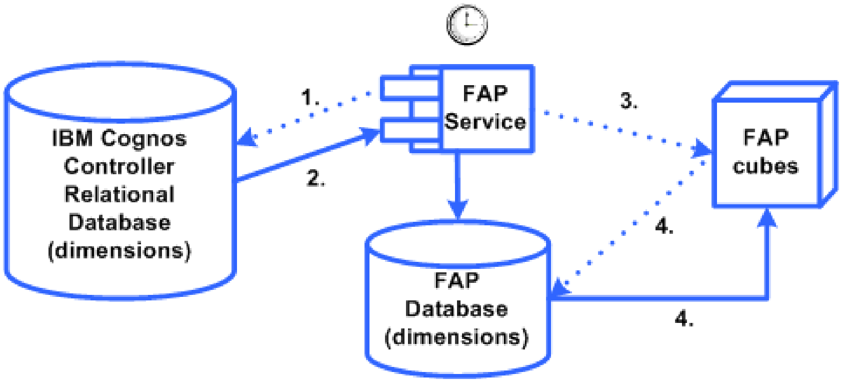
Structure updates

The initial publish process moves all dimensions as they currently appear in Cognos Controller. As it is possible to store measures on aggregation nodes in Cognos Controller, but not in TM1, the structure update process needs to manage both aggregation nodes as well as dimension changes. For more information, see "Dimension transformation principles" (p. 24).

The structure updates process contains of two parts:

- Publish children - structures are automatically moved to TM1 for the fact trickle to work.

- Update structures - the TM1 dimensions are re-generated. This is done manually by the user in the FAP client.


IBM Cognos Controller Relational Database (dimensions)
1.
2.
FAP Service
3.
FAP cubes
4.
FAP Database (dimensions)
4.

Workflow for the first part of the structure updates:

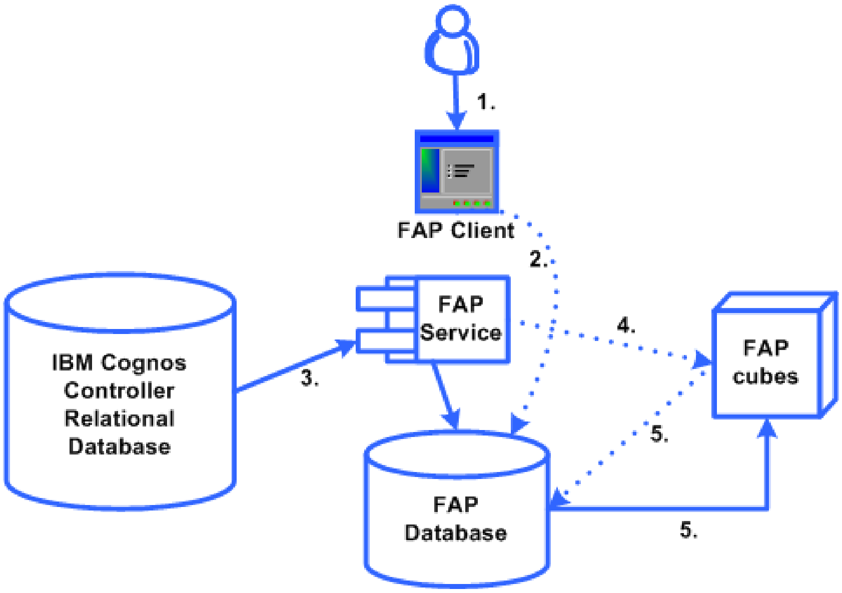
1. Based on the clock interval defined in the FAP client, the FAP Service searches for changes in the Cognos Controller database.

2. The FAP Service transfers structural changes from the Cognos Controller database to the FAP database.

3. The FAP Service calls the TM1 API and initiates a TI script.

4. Members are added in the TM1 dimensions. The cube can be reported on but is not possible to update during this operation.

Note: At this point new members are not inserted in the right place in the TM1 hierarchy. This is done in the second part of the update.



Workflow for the second part of the structure updates:

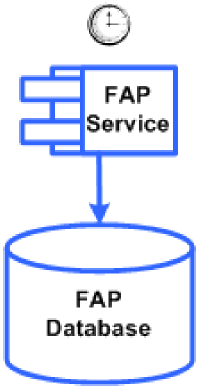
1. The user notices that there are pending dimension updates in the FAP client.
2. The user initiates the structure update in the FAP Client.
3. The FAP Service transfers structures from Cognos Controller to the FAP database.
4. A TM1 API call is triggered, to move updates from the FAP database to the FAP cubes.
5. Dimensions are updated according to the Cognos Controller hierarchy. Fact and data are kept during this operation.

Users and authorization groups are also handled in this step, which is also seen as slowly moving dimensions. Slowly moving dimensions are dimensions that change infrequently.

The update of slowly changing dimensions is a manual process due to process and technical reasons. A dimensional update could potentially invalidate reports, and even a small change in Cognos Controller could generate costly operations on the TM1 side. Typically it is beneficial to have a level of user control over this step, for example to collect a number of changes before updating the cube.

Purging the FAP database

The FAP Service also purges the FAP database. On a regular basis, the FAP Service cleans the temporary storage. This purging is done to prevent the temporary storage from getting too big over time.



The purging is based on the clock interval set in the FAP client.

FAP cubes and dimensions

For information about set-up and configuration, see the Install IBM Cognos Controller Financial Analytics Publisher chapter in the *IBM Cognos Controller Installation and Configuration Guide*.

FAP cubes

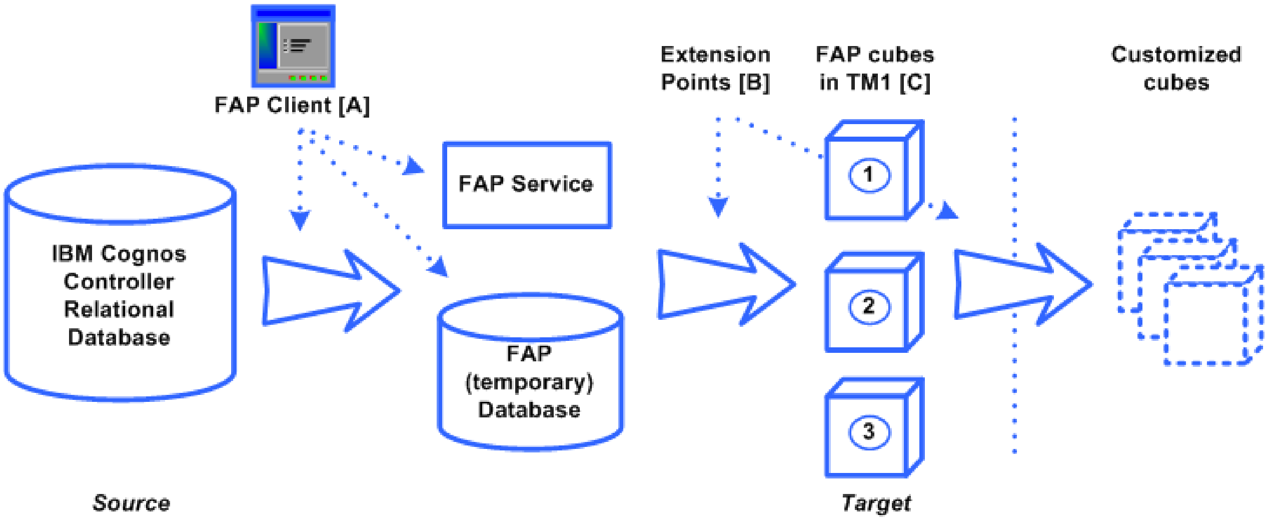
Depending on the settings in the FAP client [A in the picture], as well as the configuration in Cognos Controller, you can create up to three different FAP cubes [C in the picture]:

1. A monthly detailed cube holding leaf level data for all companies as they appear in Cognos Controller (named <Data mart name>MonthlyCompanyDetails, where you can configure <Data mart name>).

2. A monthly consolidated cube holding consolidated values from the last consolidation in Cognos Controller on group level only (named <Data mart name>ConsolidatedGroups).

3. A weekly actualities cube holding leaf level data for all companies as they appear in Cognos Controller (<Data mart name>WeeklyCompanyDetails).

   Note: Only applicable to installations with weekly actualities.



FAP cubes should not be modified because they are created or re-created in each initial publish. Hence, it is expected that additional customized cubes are created in implementation projects; cubes that might be compressed regarding dimensions and/or contain additional rules for financial logic.

The extension points [B in the picture], provide means to automate the customization of the FAP cubes through standard TM1 functionality. The FAP service will look for and execute TI scripts following a certain naming convention. For more information, see "FAP extension points" (p. 39).

Transformation rules

A number of transformation rules are applied when data is moved from the Cognos Controller relational data source to the cube target.

The transformation rules are applied for the following reasons:

- There are differences in the way that a relational database and a cube handle data For example, the Cognos Controller relational database stores values on an aggregated level whereas the cube only stores values on a leaf level.
- Cognos Controller works with time-dependent dimensions (the company dimension).

When data is changed in Cognos Controller (the source), the service moves the changes to the cube automatically.While moving the data, the service transforms it into pre-defined cube models according to a set of pre-defined rules. This transformation is handled by the FAP Service that is leveraging its own temporary database to complete its job. For a detailed description of the transformation principles, see "Dimension transformation principles" (p. 24).

Dimension naming conventions

The following naming conventions are used in the transformation. In this section, the target is the FAP cubes, and the source is the Cognos Controller relational database.

	Cube dimension (target)	Relational counterpart (source)	Monthly cube	Consol cube	Weekly cube
1	Account	Account	X	X	X
2	Company	Company	X		X
3	CompanyGroup	Company		X	
4	<Ext Dim 1>	<Ext Dim 1>	X	X	X
5	<Ext Dim 2>	<Ext Dim 2>	X	X	X
6	<Ext Dim 3>	<Ext Dim 3>	X	X	X
7	<Ext Dim 4>	<Ext Dim 4>	X	X	X
8	ActualityMonthly	PerAct	X	X	

	Cube dimension (target)	Relational counterpart (source)	Monthly cube	Consol cube	Weekly cube
9	ActualityWeekly	PerAct			X
10	PeriodMonthly	PerAct	X	X	
11	PeriodWeekly	PerAct			X
12	Currency	Currency Code	X	X	X
13	Transaction Currency	Transaction Currency Code	X	X	X
14	Consolidation Perspective	Consolidation Type	X	X	X
15	Closing Version	Manual Journal Type/Closing Version	X	X	X
16	Contribution Version	Automatic Journal Type/Contribution Version	X	X	X
17	Origin Company	Original Company	X	X	X
18	Counter Company	Counter Company	X	X	X
19	Journal Number	Journal Number	X	X	X
20	Counter Dimension	Counter Dimension	X	X	X

Cube dimension (target)	Relational counterpart (source)	Monthly cube	Consol cube	Weekly cube
Measures - YTD - Transaction YTD - Monthly - Weekly - Comment - Transaction Comment - Comment Count	Amount Amount in Transaction Currency **Notes:** • Monthly and Weekly measures are calculated through rules. The weekly value is only applicable for the weekly cube. • The comment count is calculated as the summary count of all comments on leafs below in the hierarchy. • Transaction comment is for IC.			

The Account and Company dimensions both have a number of attributes. For more information, see "Dimension attributes" (p. 41).

Examples of target cubes

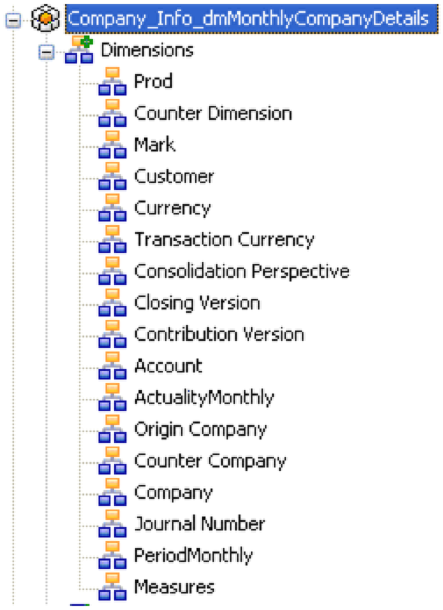
The following examples illustrate the possible target cubes. Please note that the way the naming of the extended dimensions is set-up in the source affects the cubes (in the examples these are Prod, Mark, and Customer).

The monthly detailed cube

The monthly detailed cube is the most frequently used cube. It holds facts and data and comments for all leafs defined in the Cognos Controller hierarchy. In addition, summation nodes are automatically calculated by TM1.

During the consolidation process, when a consolidation is run in Cognos Controller, the result will be consolidated values on all groups in the source. In the target cube, the same values can be seen as a result of TM1 aggregating all leaf values into the summation nodes. The consolidated values as such (in the source) are not transferred to the target. However, as soon a leaf value is updated in the source (Cognos Controller), there will be a difference on the group level when comparing the source and the target. This is because each leaf change is propagated to the target and then aggregated according to the aggregation logic in the TM1 OLAP environment. In other words, the summation nodes will contain the recent changes in the source. This is different from the source, in which the consolidation values from the last consolidation stay the same, even if a leaf value has been updated.

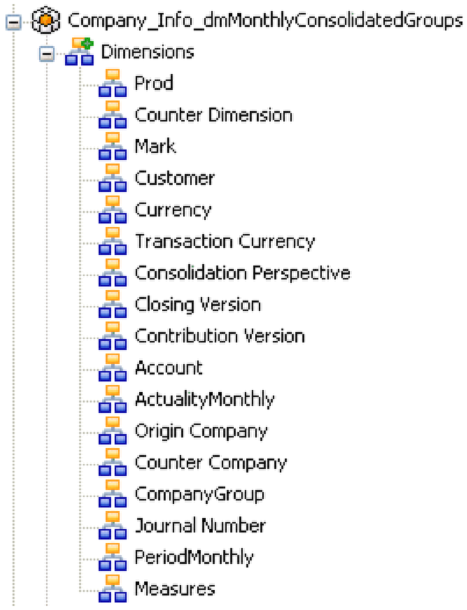
The next time a consolidation is run in Cognos Controller, the source and the target will look the same again.



- Company_Info_dmMonthlyCompanyDetails
  - Dimensions
    - Prod
    - Counter Dimension
    - Mark
    - Customer
    - Currency
    - Transaction Currency
    - Consolidation Perspective
    - Closing Version
    - Contribution Version
    - Account
    - ActualityMonthly
    - Origin Company
    - Counter Company
    - Company
    - Journal Number
    - PeriodMonthly
    - Measures

The monthly consolidated cube

The monthly consolidated cube holds the consolidated values for companies defined as groups (or sub-groups) in Cognos Controller. This type of cube is not affected by leaf-level changes in Cognos Controller. This cube will be updated only when a consolidation is run in Cognos Controller.



- Company_Info_dmMonthlyConsolidatedGroups
  - Dimensions
    - Prod
    - Counter Dimension
    - Mark
    - Customer
    - Currency
    - Transaction Currency
    - Consolidation Perspective
    - Closing Version
    - Contribution Version
    - Account
    - ActualityMonthly
    - Origin Company
    - Counter Company
    - CompanyGroup
    - Journal Number
    - PeriodMonthly
    - Measures

The consolidated cube leverages a variant of the company dimension, CompanyGroup. This is a flat hierarchy containing only the groups and without the period stamp.

By combining data from the detailed and the consolidated cubes (by customizing), it is possible to show the result of the last consolidation on group levels (summation nodes) and latest updates on company level (leaf nodes).

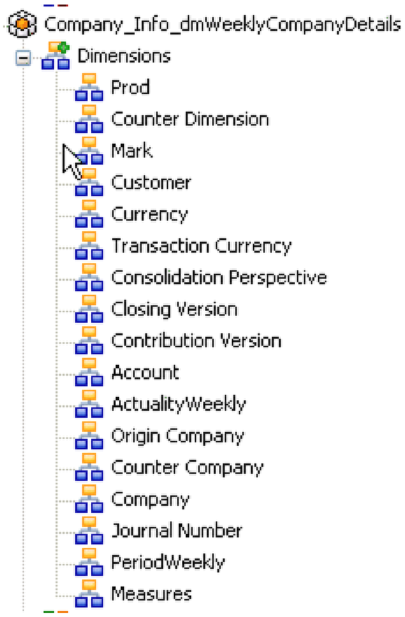
The weekly cube

The weekly cube is only applicable to Cognos Controller installations where weekly actualities have been configured.

This cube is similar to the monthly cube but with two main differences:

- It uses the ActualityWeekly (instead of the ActualityMonthly) dimension.
- It uses the PeriodWeekly (instead of the PeriodMonthly) dimension.

Please note that the hierarchy for the PeriodWeekly dimension is Year->Week; that is, month is not part of the hierarchy. This is because Cognos Controller does not handle the relationship between months and weeks; that is, determining what weeks, or parts of weeks, that go into a certain month.



Company_Info_dmWeeklyCompanyDetails
Dimensions
Prod
Counter Dimension
Mark
Customer
Currency
Transaction Currency
Consolidation Perspective
Closing Version
Contribution Version
Account
ActualityWeekly
Origin Company
Counter Company
Company
Journal Number
PeriodWeekly
Measures

Additional considerations

The sections that follow discuss some additional considerations to keep in mind when working with Cognos Controller Financial Analytics Publisher.

Near real time FAP versus published target

Cognos Controller Financial Analytics Publisher automatically and in near real time moves any change from the relational source to the target cube. This enables in-process reporting and data validation based on data in the cube.

You can give report consumers access to the final data, but not near real time cubes. This is useful for users that only require a snapshot of the data for both statutory and management reporting after the closing. There are several ways to accomplish this:

- Turn off the trickle publish function. However, this will completely eliminate the possibility for close to real time analysis for all users.

- Make a copy of the near real time cube within the same TM1 server and use the copy for these users.
- Copy the TM1 server.

For additional descriptions on how to customize FAP for various reporting scenarios, go to the Cognos Proven Practices documentation web site: http://www.ibm.com/developerworks/data/library/cognos/cognosprovenpractices.html

FAP and security

Users and authorization groups in Cognos Controller are published to TM1, and the authorization groups get prefixes to avoid naming conflicts. The security modes available for Cognos Controller and FAP in TM1 are described in the following sections:

Basic security mode

For TM1 9.4.1 and TM1 9.5.x, Cognos Controller users and authorization groups are published and can be leveraged if CAM authentication is not used to access the Cognos Controller Financial Analytics Publisher cube (for example from the TM1 Excel plug-in, but not from BI).

Note: Cognos Controller users and authorization groups present in Cognos Controller are deleted in TM1 during the initial publish operation.

CAM security mode

For TM1 9.4.1, all CAM users in Cognos Controller will be published but without the integrated security from Cognos Controller that is present in the TM1 cube.

For TM1 9.5 and later, there is integrated security between Cognos Controller and TM1. This means that users and authorization groups in Cognos Controller are published to TM1. Then for all CAM users present in TM1, the CAM user ID will be connected to the Cognos Controller user ID (provided the CAM information has been maintained in Cognos Controller) and get the appropriate authorization groups.

TM1 Security Mode Settings that are not supported by Cognos Controller will result in the initial publish process being aborted and the datamart being set to Error. The following TM1 API security modes are not supported:

- Distributed

  Implies that the TM1 server is a distributed server that accepts connections without specifying any credentials.

- Mixed

  Implies that the TM1 server accepts user authenticating either using Basic authentication or Windows Integrated Authentication.

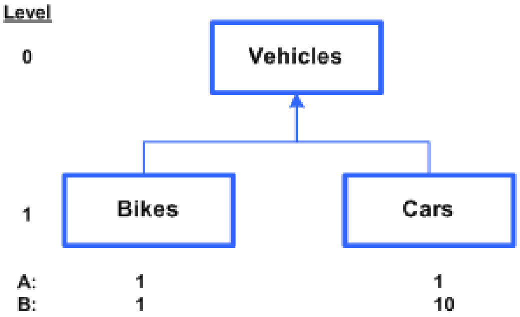
- WIA

  Implies that the TM1 server accepts connections that can authenticate based on Windows Integrated Authentication.

Aggregations (statistical accounts)

When working with statistical accounts, you need to be aware of the numerator-denominator behavior as described in the following example:

Imagine a simple product hierarchy with Vehicles as the top node (0) and with Bikes and Cars as the next level (1). In this example, a certain calculation (KPI) is defined as C = A / B, where A is the numerator and B the denominator. C is to be calculated on both Level 0 and Level 1.



Calculating C with the support of the advanced formula calculation accounts could be done in two different ways:

- C is set up as an advanced formula calculation account at Level 1, and the division is part of the formula. A and B could be normal accounts or AFC/statistical accounts.
- C is calculated in the report. A and/or B are set up as advanced formula calculation/statistical accounts at level 1.

From a calculation point of view:

Formula	Results
CBikes = ABikes / BBikes	CBikes = 1/1 = 1
CCars = ACars / BCars	CCars = 1/10 = 0.1

If C is set up as an advanced formula calculation account, the FAP cube would automatically aggregate C to the next dimension level (Level 0). However, CVehicles is not defined as CBikes + CCars, but rather as CVehicles = AVehicles / BVehicles = (ABikes + ACars) / (BBikes + BCars.

Default aggregation for C on Level 0:

- CVehicles = CBikes + CCars = 1 + 0.1 = 1.1

Expected result for C on Level 0:

- CVehicles = AVehicles / BVehicles = (ABikes + ACars) / (BBikes + BCars = (1+1) / (1+10) = 2/11 = 0.1818

When defining these types of calculations you must be aware of this behavior. However, there are at least two possible alternative ways of handling this:

- Scenario 1: If you want the calculation of C to be set up as an advanced formula calculation account, then different advanced formula calculation accounts should be created by dimension level. For example, C dimension level 0 and C dimension level 1. You then need to use the appropriate advanced formula calculation account in your reports; that is, you need to be aware of that C on level 1 will be aggregated to Level 0, and not be meaningful on this level.

- Scenario 2: If the calculation of C is done in the report for all levels, no such aggregation would occur. A and/or B can be set up as advanced formula calculation accounts, use the built-in formulas and period logic, and be re-used in multiple KPI calculations in various reports.

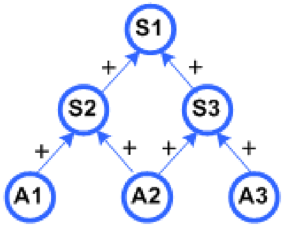
Please note that an advanced formula calculation performed on the top level only would work without the numerator-denominator consideration.

Dimension transformation principles

To understand the FAP cubes and how to author reports based on data in the cubes, it is important to understand the underlying transformation principles.

The Account dimension

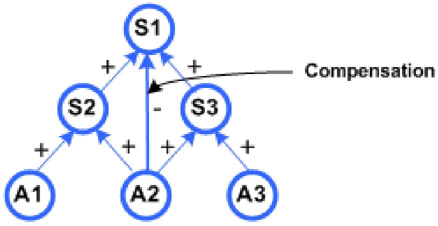
The account structure in Cognos Controller's relational source allows a base account to logically sum-up several times to the same summation account, as illustrated in the following example.



From a hierarchical point of view, S1 would be evaluated as: S1 = S2 + S3 = (A1+A2) + (A2+A3) = A1 + 2*A2 + A3.

However, the correct interpretation is that a single account should not be summed multiple times, hence: S1 = A1 + A2 + A3.

To avoid flattening the account structure and losing the ability to drill down in the target, a compensating transformation, such as the following is used:



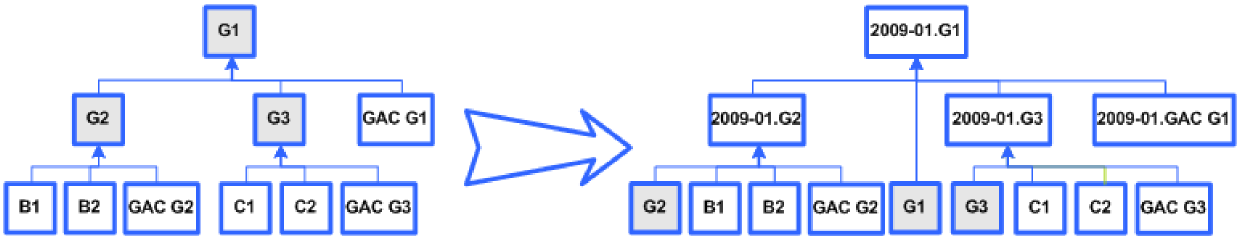
In this compensating transformation, the summation structure in the source is rebuilt in the target and includes relations that are weighted to compensate where needed, for example 2* A2 - A2 = A2.

Note: In the account structure there is no need for additional aggregation nodes because summation accounts do not relate to any physical transactions.

The Company dimension

The company structure is period dependent in the source; one structure version per month. IBM Cognos Controller Financial Analytics Publisher can handle multiple company structures, for different consolidation types and periods. This allows you to drill down through the group and company tree for other periods.

The transformation principle is illustrated in the following example. To the left, the Legal (LE) company structure is in the source, and to the right, the target is. In the target there is a leaf level of companies and group entities (G1, G2 and G3) that relate to the transactions (elimination companies). In the illustration, the Group Adjustment Company (GAC) in the source is treated like any other company in the hierarchy.



Summary of the transformation principle:

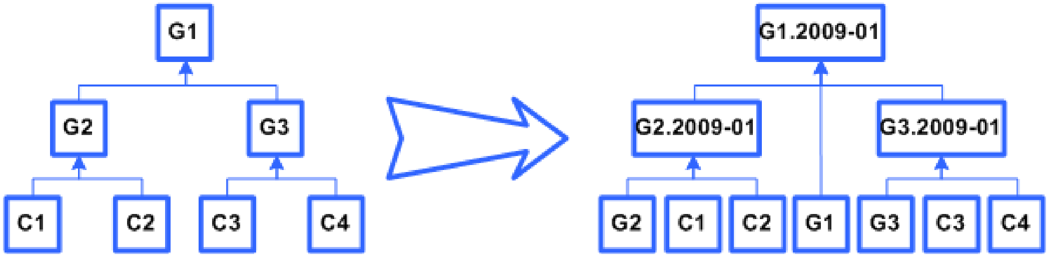
- The names (identifiers) of the leaf nodes are invariant over different periods and the names in the target cube correspond to the codes in the source.
- Each published period (that is, the period that the user chooses) has its own aggregation hierarchy. The naming convention contains both the corresponding source group code and the period.
- The company dimension in the target is the sum of the period specific hierarchies illustrated and the leaf nodes (corresponding to entities in the source). This means that, in the target, the company dimension will have one top node per period.
- The naming convention is <consolidation type>.<period>.<company>, for example, LE.2009-09.G2

The principle is similar to the one used for extended dimensions. Each "transaction carrying member" becomes a leaf node in the target. Each leaf node has the same name as the code in the source, which simplifies updates. The aggregating nodes in the target are added and follow a naming convention.

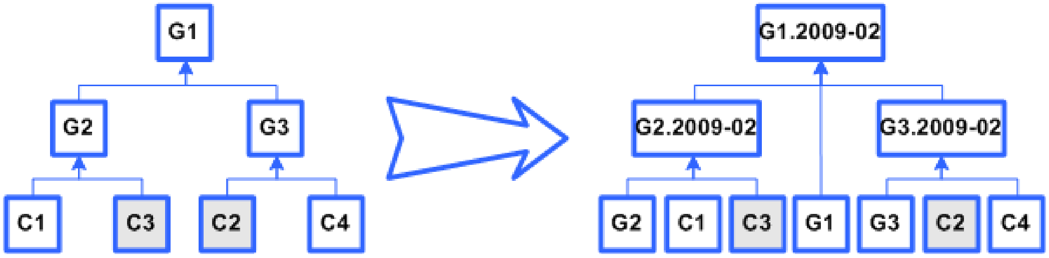
Example of the time dependent company dimension

The company dimension is time dependent. That is, it can vary over time, or more specifically, it varies with the period. A new company might be added (acquired), another might be removed (sold), and a third might be moved in the structure (ownership change).

The following example describes the transformation of a legal company structure as it looks in the 2009-01 period.



In the next period, the company structure is modified and company C3 is now owned by G2 and C2 by G3. The new transformation would look like this.



Each published period (that is, each period that the user chooses when setting up the trickle-publish) has its own aggregation hierarchy. The naming convention contains both the consolidation type, the group code, and the period from the source.

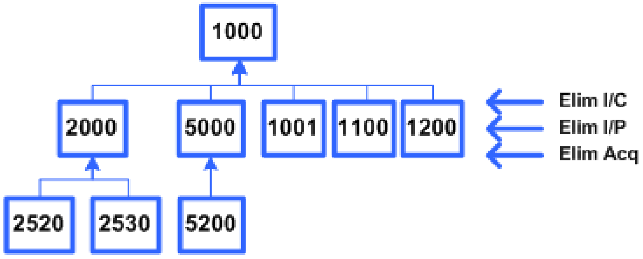
The company dimension in the target is the sum of the period specific hierarchies illustrated and the leaf nodes. This means that the target company dimension will have one top node per period.

In the current solution, it is possible to drill down through the company hierarchy for one period at a time. However, if you drill down through the 2009-01 company hierarchy after having selected the 2009-03 period dimension, the result will be inconsistent. Time-dependent dimension aggregation is not currently supported.

Comparison of the trial balance in Cognos Controller Financial Analytics Publisher and the Publish to Data Mart function in Cognos Controller

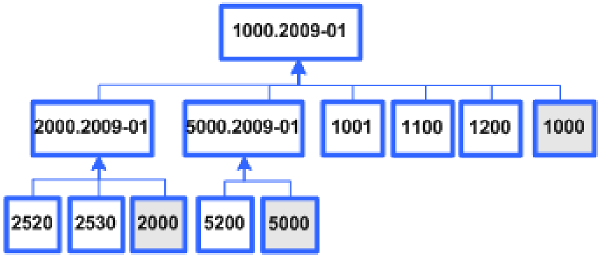
The following example illustrates the difference between the trial balance in IBM Cognos Controller Financial Analytics Publisher and the Publish to Data Mart function in IBM Cognos Controller.

Source



In the trial balance report for the 0901 period, actuality AC, and the company structure 0901LE, the eliminations (AJT) will appear as separate columns for each company when you select the option in the report (this option is default in each company and drillable). On the subgroup, these eliminations from the subsidiaries will be aggregated and kept as such (except if you select the option Consolidated Automatic journals type as BASE). When drilling by automatic journal type, you will see eliminations that are directly on the subgroup level.

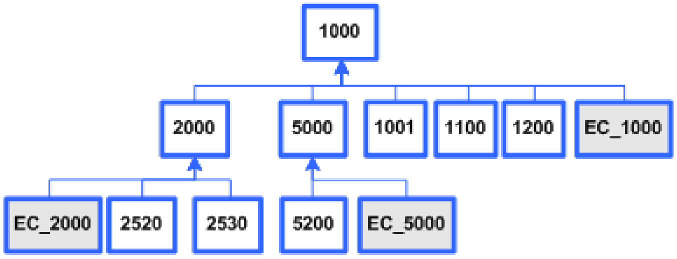
Target



When the cube is generated for the company structure version 0901LE, period 0901, and actuality AC; the Elim I/C, Elim I/P and Elim Acq transactions will all go to the elimination company 1000, assuming they were booked on sub-groups 2000 and 5000. Eliminations in the source that are stored in companies that are not cross-owned will remain on the company in the target, please also review the server preference ETYP_EXCLUDE. For more information, see "Contribution versions" (p. 32).

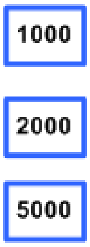
The Publish to Data Mart function in Cognos Controller

If you want to use the **Publish to Data Mart** function in Cognos Controller instead, the transformation will look like the following diagram. Note that the **Publish to Data Mart** function does not support the time dependent dimension concept of Cognos Controller Financial Analytics Publisher, and that the elimination company naming convention is slightly different.



The CompanyGroup dimension

The CompanyGroup dimension contains all groups in a flat hierarchy and without period stamps. This dimension is used in the consolidated cube only.



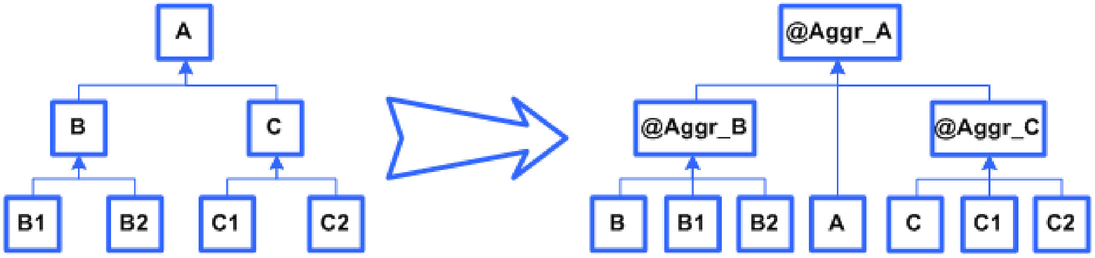
Extended dimensions 1-4

The same names are used in the cube as in the relational database. In the target, an extended dimension is built according to the following conventions:

- All members in the source become leaf members in the target, and use the same code.

- All members that are aggregation members in the source are also created as separate aggregation nodes in the target.

The difference between the cube and the relational source is that, in the source, values are stored on aggregated levels in the hierarchies. This is not possible in the cube, and aggregation nodes are added to solve this.



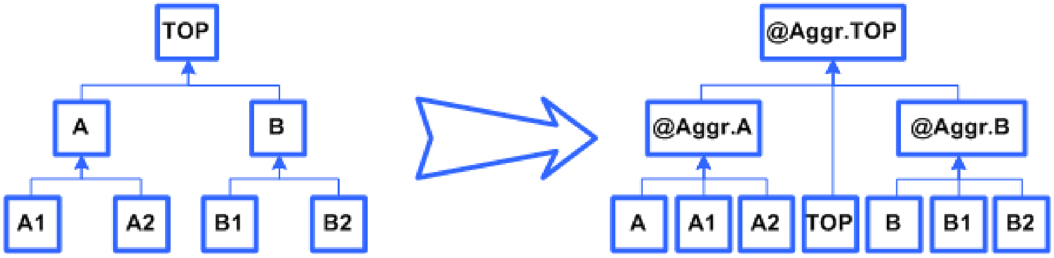
Example of slowly moving dimensions

Slowly moving dimensions are dimensions that are not frequently updated.

Slowly moving dimensions are updated in two phases:

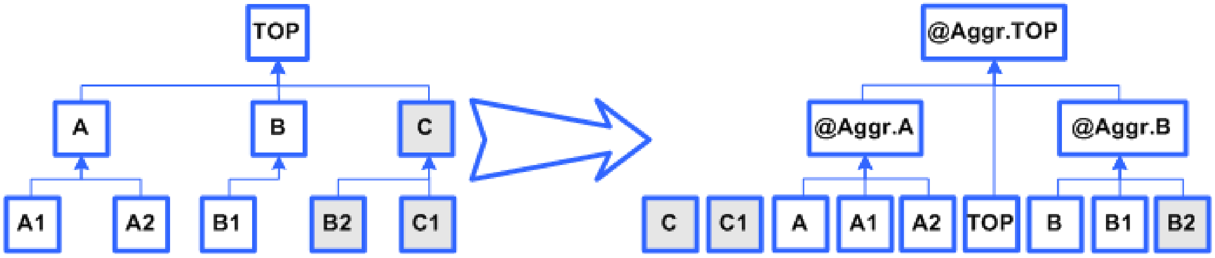
- The first phase handles transactions in the source with members that are unknown to the target. For example, this is typically the result of an added product, company, or period in the source.
- The second phase updates one or many dimensions in the target.

In the example, the source is to the left and target after transformation is to the right. Please note that the naming convention @Aggr for the system created aggregation nodes can be changed in the FAP client.



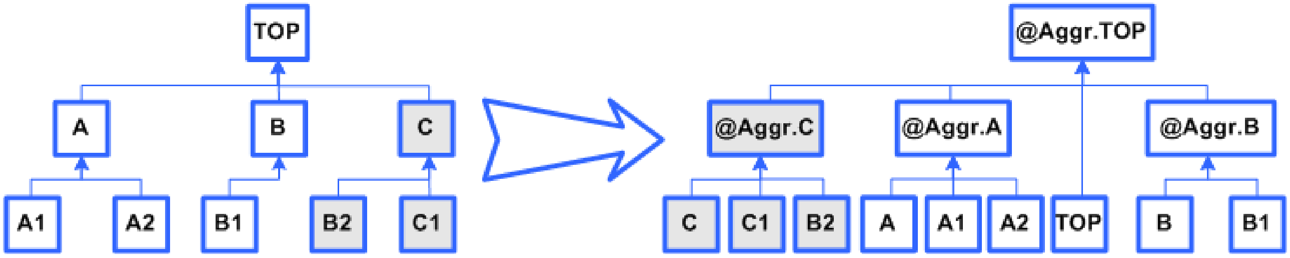
A1, A2, B1, and B2 could, for example, be different products, belonging to different product groups. Product group C and product C1 are added and product B2 is moved.

Initially, the change in the source is not reflected in the target. However, after transactions corresponding to the new product members (C and C1) are detected, the new members are added to the target as shown in the following example:



These new members are not connected in the hierarchy. For convenience, the new members are collected under a "not connected" node. The only dimensional operation that occurs during this stage is to add the new leaf members. The Cognos Controller Financial Analytics Publisher server keeps track of all valid leaf members in all dimensions to detect the new ones. A new leaf node is added before the corresponding transaction is imported into the target.

Over time, new members are added, and there will be inconsistency that must be addressed. This is the second phase of the slowly moving dimension transformation. In the Cognos Controller Financial Analytics Publisher client it is possible to detect that there are pending source dimension changes. The user then has the option to update the TM1 dimension to be consistent with the source. This update does not only contain new members and a restructured aggregation hierarchy, it also involves changes of names (aliases on the target side). This is what it will look like after the user manually triggers the dimensional updating:



The user needs to update the dimensions manually for the following reasons:

- It is not specified how an automated detection of a dimension update in the target should be updated from the source
- There is a disadvantage with automatic updating because every small change in the source will result in a costly operation in the target. For example, it is best to not trigger an update after each change if a user is making significant changes to the product hierarchy.
- Even if performance is not a concern, the user might not want the update to be triggered automatically. A dimensional update could potentially invalidate reports and it might be best to wait until a certain time, for example after the closing process is finished.

The ActualityMonthly dimension

The actualitymonthly dimension is a list of the used actualities without any hierarchical logic. The actuality is transformed as it is logically used in the relational source.

In the source fact table, actuality is not a column of its own, that is, the PerAct dimension is split into two dimensions in the target: actuality and period. Furthermore, the period and actuality dimensions are sometimes treated as one dimension in the source, and in other cases as two different dimensions.

Linked actualities

Linked actualities are also moved to the ActualityMonthly dimension.

Note: A linked actuality increases the size of the cube, because it is a copy and not a reference.

The ActualityWeekly dimension

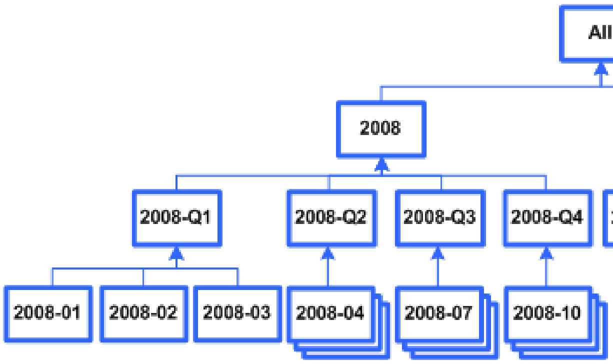
The ActualityWeekly dimension is, in the same way as the ActualityMonthly dimension, a list of actualities without any hierarchical logic. The actuality is transformed as it is logically used in the relational source. However, only source actualities defined as weekly are included in this dimension, and they are used in the weekly cube only.

The PeriodMonthly dimension

Similar to actualitymonthly dimension, periodmonthly is a separate dimension and is aggregated from months into quarters and then into years. The periodmonthly dimension's year and quarter levels are not explicitly present in the source. The period hierarchy is not meant for aggregation; its purpose is to support relative period logic and to support relative periods in the year following the financial year, and not the calendar year.

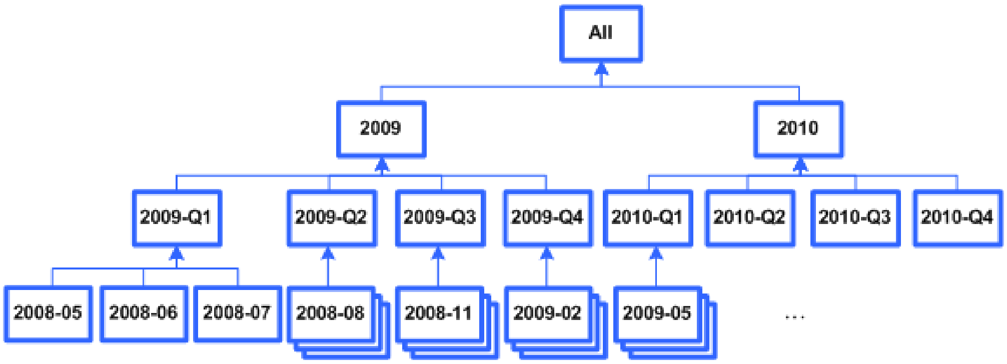
Example of closing period in December

Note that the node All is not explicitly present in the periodmonthly hierarchy, but is used in this example to make it easier to understand.



Example of closing period in April

April is used as the closing month in this example.



Comments:

- The months that belong to the same year are calculated based on a closing period.
- The naming convention used most often for broken financial years is the "year of the closing month". In the data mart, the year always relates to the financial year and not to the calendar year.

- The naming convention for the quarter level in the period dimension follows from the years. For example, the financial year 2009 will have four (financial) quarters named 2009-Q1 to 2009-Q4. The quarters must be uniquely named, so you cannot use just Q1 for all years.

One important objective of the period dimension is to support the notion of relative periods, for more information see "Example of relative periods" (p. 31).

Example of relative periods

A typical report will list the profit and loss statement for one company but for different periods. For example, a table report might have accounts as rows and different periods as columns. The period columns are all based on one current period which the report consumer will be prompted for. For example, the columns could be "current period", "previous period", "last year's closing period" and "last year's corresponding period". In the Report Generator in IBM Cognos Controller, the three periods would be expressed as "-1", "-N112" and "-12"; all relative to the prompted period ([currentperiod]).

In the target, these relative period functions are not supported, but the report author can instead use the query functions available in Report Studio and other reporting products.

Source expression	Report Studio query function (MDX)
-1	lag([currentperiod],1)
-12	lag([currentperiod],12)
+1	lead([currentperiod],1)
+12	lead([currentperiod],12)
P001	firstSibling([currentperiod])
P012	lastSibling([currentperiod])
N101	firstChild (lag(parent([currentperiod]),1))
N112	lastChild (lag(parent([currentperiod]),1))

Note that the BI syntax uses navigation in the period hierarchy.

The PeriodWeekly dimension

The PeriodWeekly dimension is similar to PeriodMonthly, however, it is aggregated from weeks into years. It is used in the weekly cube.

The Currency dimension

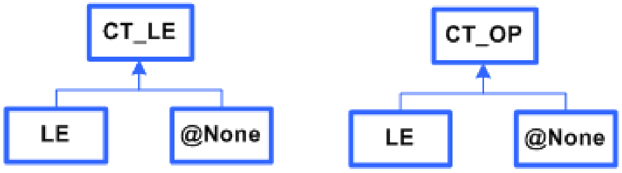
The currency dimension is transformed in a straightforward 1:1 fashion, that is, the target looks the same as the source.

The Transaction currency dimension

The transaction currency dimension is transformed in a straightforward 1:1 fashion, that is, the target looks the same as the source.

Consolidation perspective

The transformation principle for the consolidation perspective is straightforward. Reported values and manual journals (ktypkonc is blank) get no consolidation perspective. Other records, that is, automatic journals get the consolidation perspective as the consolidation type, for example, LE and CT_LE are an aggregated node.



Automatic Journals can be booked on companies or on groups, which means that the consolidation perspective filter affects values on both levels.

Please note that automatic journals may or may not include a company reference, depending on whether there is cross-ownership involved (a company reference) or not (no company reference). For more information, see "Contribution versions" (p. 32).

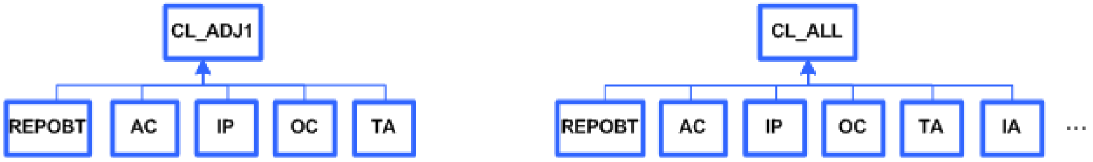
Closing versions

The manual journal types with the closing version aggregation level in the source are called closing versions in the target, and have a 1:1 transformation rule.

The closing versions in the target are prefixed with CL. Manual journal types have the same code as in the source with one exception; the reported value journal type, which is displayed as REPOBT because ' ' is not a valid name in the target.

Example of closing version

Note that the code for a reported value (blank in the source) is named REPOBT in the target. This is to avoid blank names and guarantee unique names.



Contribution versions

The contribution versions in the target are prefixed with CO. Automatic journal types have the same code as in the source with one exception; the base value automatic journal type which is displayed as BASEET because ' ' is not a valid name in the target. You can configure the naming convention in the FAP client.

The transformation of automatic journals introduces complexity due to support for cross-ownership in Cognos Controller. The challenge is to store values in a summation node in a cube while accommodating the way in which Cognos Controller handles cross-owned companies.

For this transformation the concept of elimination company (EC) is used, which is similar to the Publish to Data Mart function. In Financial Analytics Publisher, the naming convention is to give the same name for the EC company as for the group. For example, for the group named Grp1, the elimination company is also called Grp1. For more information, see "The Company dimension" (p. 25). In the Publish to Data Mart function in Cognos Controller the name would have been EC_Grp1.

On a high level, all automatic journals 30 (rebooking of subsidiaries change in equity) and automatic journal 37 (allocations) are by default kept on the company level in the target. Other automatic journals are moved to the elimination company in the target. A server preference (ETYPE_EXCLUDE) is available to modify this default behavior. By using ETYPE_EXCLUDE you state which automatic journal types should remain on the company in the target, instead of being transferred to the elimination company. Note that this does not affect eliminations on groups or cross-owned companies; those journals will always be transferred to the elimination company.

Note: Any modification of the server preference should be done with caution, especially if cross-ownership exists in the company structure.

It is followed by these rules:

- When company type = K (that is, the sub-group) and automatic journal type = " " (that is, the automatic journal is blank) then exclude the row completely from the transformation. This is because these journals are also in a company below (and we do not want them to be stored twice).

Followed by these set of rules:

1. Transactions (automatic journals) with specific KtypKonc (cross-ownership) move to the EC company.

   For example, in the case of cross-ownership, the ktypkonc would be LEGrp1, and not LE. In situations without cross-ownership, the ktypkonc would be LE or OP.

2. Transactions with KtypKonc <> " " on sub-groups (that is, automatic journals booked directly on sub-groups) move to the EC company belonging to the group above.

3. Transactions with KtypKonc = " " - stay on the company (BASE values), unless the server preference says differently.

4. Automatic journal types "30" and "37" stay on the company, unless the server preference is set and excluding these two.

   Server preference addition:

   If the server preference is set, automatic journal types in the server preference stay on the company, all other automatic journal types go to the EC company:

   - For example, if you want journal types 30, 35, 37 and 38 to stay at the company, the server preference should look like this: "ETYPE_EXCLUDE=30, 35, 37, 38" (that is, you need to mention 30 and 37 despite the default behaviour of storing at company level).

Example of contribution version

The following table provides an example of a transaction in the source:

Comp	Automatic Journal	kTypCons	Period	...
A	90	LEGrp2	2007-01	

This will be transformed into:

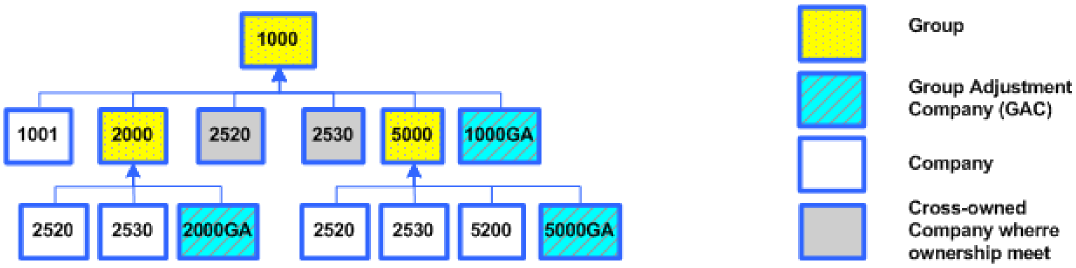
Comp	Automatic Journal	origComp	period
Grp2	90	A	2007-01

In this example, Grp2 is the elimination company of the target group.

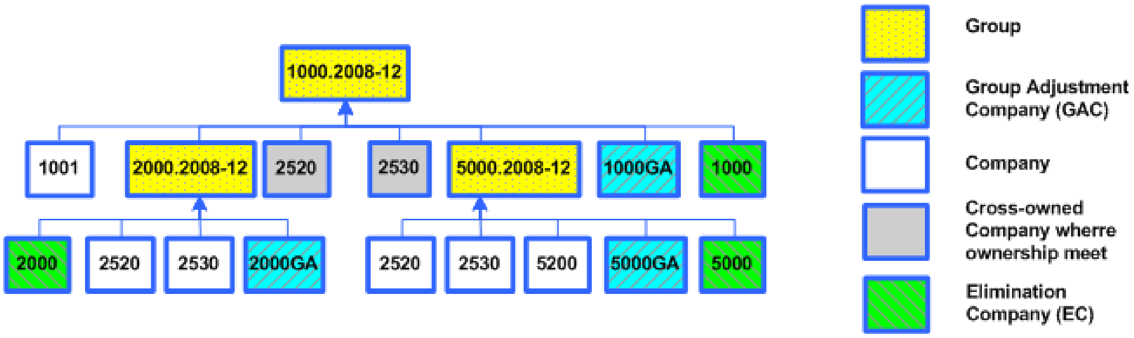
Examples of contribution version - trial balance

This example shows company codes in both the source and the target and how data is transferred following the different contribution version rules.

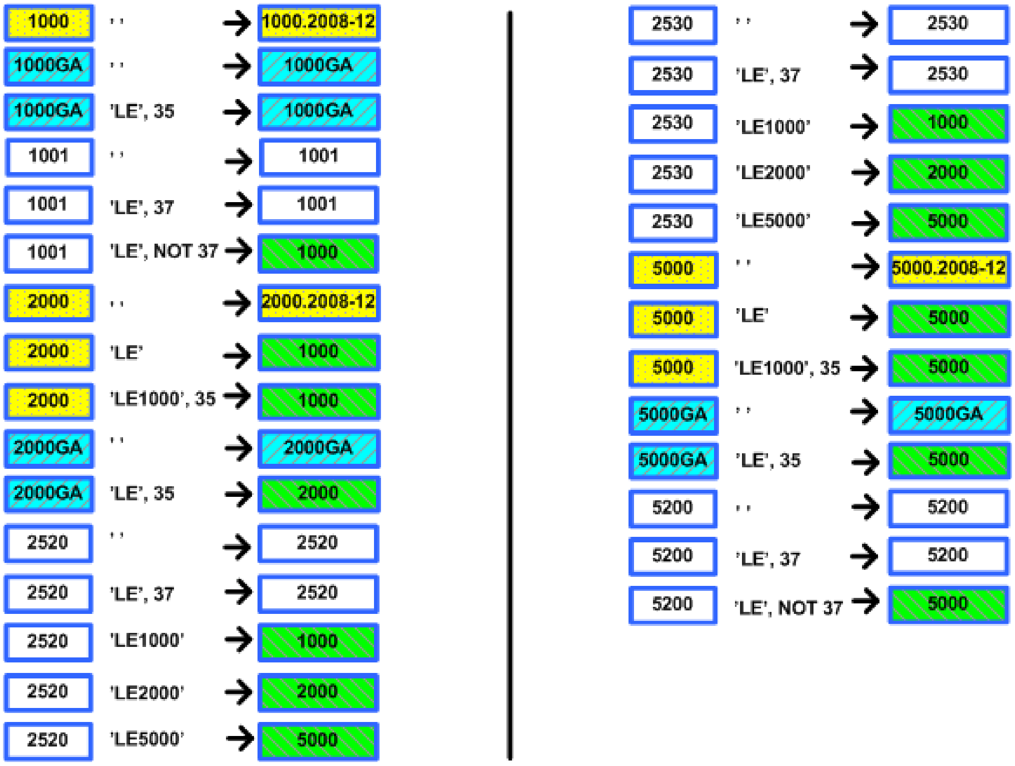
Source:



Target:



The following example illustrates the Cognos Controller trial balance for period 0812, actuality AC, and company structure 0812LE, and the transformation to the target cube for period 0812, actuality AC, and company structure 0812LE.



1000	' '	→	1000.2008-12
1000GA	' '	→	1000GA
1000GA	'LE', 35	→	1000GA
1001	' '	→	1001
1001	'LE', 37	→	1001
1001	'LE', NOT 37	→	1000
2000	' '	→	2000.2008-12
2000	'LE'	→	1000
2000	'LE1000', 35	→	1000
2000GA	' '	→	2000GA
2000GA	'LE', 35	→	2000
2520	' '	→	2520
2520	'LE', 37	→	2520
2520	'LE1000'	→	1000
2520	'LE2000'	→	2000
2520	'LE5000'	→	5000
2530	' '	→	2530
2530	'LE', 37	→	2530
2530	'LE1000'	→	1000
2530	'LE2000'	→	2000
2530	'LE5000'	→	5000
5000	' '	→	5000.2008-12
5000	'LE'	→	5000
5000	'LE1000', 35	→	5000
5000GA	' '	→	5000GA
5000GA	'LE', 35	→	5000
5200	' '	→	5200
5200	'LE', 37	→	5200
5200	'LE', NOT 37	→	5000

The Origin company dimension

The origin company dimension (source company) contains all companies in the source as members in a simple structure.

The Counter company dimension

The counter company dimension (source company) contains all companies in the source as members in a simple structure.

The Journal number dimension

The journal number dimension is transformed 1:1, that is, the target looks the same as the source.

The Counter dimension

The counter dimension is a duplicate of the external dimension 1, and uses the same transformation principle.

The Measurement dimension

In the cube there is a measurement dimension, which corresponds to the two value columns in the fact table; amount and transamount. In the target this dimension will have three members, with the option to add more in the future.

The members are as follows:

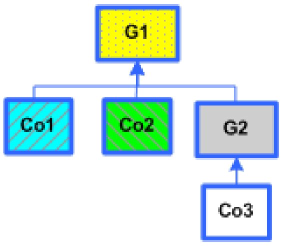
- YTD - corresponds to the amount column in the XDBxx table. This is a year to date (YTD) value in the source.

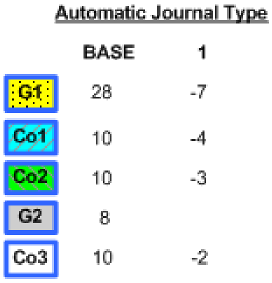
- Transaction YTD - corresponds to the transamount column.
- Monthly - this is a computed period value. For the first period in a year: the period value = the YTD value. For all other periods: period value (period) = YTD (period) - (YTD (period-1). This calculation is implemented as a cube rule.
- Weekly - similar to monthly, this is also a computed value.
- Comment - for accounts (not journals).
- Transaction Comment - for IC transactions (likely always empty).
- Comment Count - this is a computed value created as the summary of all comments on leafs below. If there is a comment or transaction comment on a leaf, the comment count gets the value of 1.

The Impact of contribution version BASE value in Cognos Controller, versions 8.x

In Cognos Controller, versions 8.x, the AJT is aggregated to BASE at subgroup levels, which causes a difference between the source and the target cube. From Cognos Controller, version 10.1 and later, this is not the case as AJT is not aggregated to BASE at subgroup levels anymore.)For subgroups, the target will show aggregated data from all subsidiary companies for the contribution version BASE. In the source, a report on subgroups will show BASE values for all subsidiary companies including eliminations that are also summarized in Base.

The following picture shows a simple structure for consolidation type LE in the source, including some data:




Automatic Journal Type

	BASE	1
G1	28	-7
Co1	10	-4
Co2	10	-3
G2	8	
Co3	10	-2

In the source, the resulting trial balance report on the highest group level (G1), for contribution version BASE, will appear as shown in the following table:

	Co1	Co2	G2	Tot	Consolidated Value
Account 1	10	10	8	28	21

	Co1	Co2	G2	Tot	Consolidated Value
Explanation			10-2 Consolidated BASE value on G2	10+10+8	28-3-4 Consolidated BASE value on G1

In the source, the resulting trial balance report on the subgroup level (G2), for contribution version BASE, will appear as shown in the following table:

	Co3	Tot	Consolidated Value
Account 1	10	10	8
Explanation			10-2 Consolidated BASE value on G2

In the target cube, for contribution version BASE, it will appear as shown in the following table:

	G1	Co1	Co2	G2	Co3
Account 1	30	10	10	10	10
Explanation	Aggregated BASE value from Co1, Co2 and Co3.			Aggregated BASE value from Co3.	

In the source, the trial balance report on the total level (including eliminations, for example BASE + automatic journal type 1), will appear as:

	Co1	Co2	G2	Tot	Consolidated Value
Account 1	6	7	8	21	21
Explanation	10-4	10-3	10-2 Consolidated value on G2	10-4+10-3+8	6+7+8 Consolidated value on G1

In the target cube on the total level (including eliminations, for example BASE + automatic journal type 1), it will appear as:

	G1	Co1	Co2	G2	Co3
Account 1	21	6	7	8	8
Explanation	Aggregated value from Co1, Co2 and Co3			Aggregated value from Co3.	

Appendix A: FAP extension points

TM1 as a platform opens up for endless possibilities when it comes to customizing the default FAP cubes, that is, to incorporate specific customer requirements into the reporting solution.

One available TM1 mechanism is the TI scripts (Turbo Integrator). For more information, see the TM1 user documentation.

FAP provides several script place holders, that is, distinct places in the code where the FAP Service will look for Turbo Integrator scripts on the TM1 server and if they follow a certain naming convention execute them. Available place holders are:

Extension point	Description
ccr_ip_before_[a-zAZ0-9]	Before initial publish.
ccr_ip_middle_[a-zAZ0-9]	In the middle of initial publish (after dimensions/structures, before data).
ccr_ip_after_[a-zAZ0-9]	After initial publish.
ccr_pc_before_[a-zAZ0-9]	Before publish children*.
ccr_pc_after_[a-zAZ0-9]	After publish children.
ccr_us_before_[a-zAZ0-9]	Before update structures**
ccr_us_after_[a-zAZ0-9]	After update structures.
ccr_dt_before_[a-zAZ0-9]	Before data trickle.
ccr_dt_after_[a-zAZ0-9]	After data trickle.

* Step 1 for slowly moving dimensions (move dimensions to TM1 for data trickle to work).

** Step 2 for slowly moving dimensions (re-generate the TM1 dimensions).

Several scripts can be executed at each step. They are then run in alphanumeric order, for example, A1 runs before A2. Although this document is not intended to cover possible TI scripts a couple of examples are provided below.

This script would add an "All"node to the company hierarchy.

```
While (j <= Dimsiz('Company'));
  Element = DIMNM('Company', j);
  If (DTYPE('Company', Element) @= 'C');
    # All should not be included into itself.
    If (Element @<> 'All');
      # Consolidation types should not be included. They are always
```

Licensed Materials – Property of IBM

```
of size 2.
      If (Long(Element) <> 2);
        # Consolidation type + period should not be included. They
are always of size 2+1+7=10.
        If (Long(Element) <> 10);
          DimensionElementComponentAdd('Company', 'All', Element,
0);
        EndIf;
      EndIf;
    Else;
       allIindex=j;
    EndIf;
  EndIf;
  j = j + 1;
End;
```

This example describes how an attribute can be added to the Company dimension, with just the code from Controller.

```
# Add one extra attribute: 'Company Code' containing
the controller code,
# except QAR companies, which gets their prefix.
AttrInsert('Company', 'Local Currency', 'Company Code', 'S');
i = 1;
While (i <= Dimsiz('Company'));
  Element = DIMNM('Company', i);
  If (DTYPE('Company', Element) @= 'C');
    # Consolidated nodes
    CodeLength = Long(Element);
    If (CodeLength > 10);
      CodeName = SUBST(Element, 12, CodeLength - 10);
      AttrPutS(CodeName , 'Company', Element, 'Company Code');
      # Copy attributes from leaf to consolidated node.
      EN_longName = ATTRS('Company', CodeName, 'EN_long');
      AttrPutS(EN_longName, 'Company', Element, 'EN_long');
      EN_shortName = ATTRS('Company', CodeName, 'EN_short');
      AttrPutS(EN_shortName, 'Company', Element, 'EN_short');
      SV_longName = ATTRS('Company', CodeName, 'SV_long');
      AttrPutS(SV_longName, 'Company', Element, 'SV_long');
      SV_shortName = ATTRS('Company', CodeName, 'SV_short');
      AttrPutS(SV_shortName, 'Company', Element, 'SV_short');
    EndIf;
  ElseIf (DTYPE('Company', Element) @<> 'C');
    AttrPutS(Element, 'Company', Element, 'Company Code');
  EndIf;
  i = i + 1;
End;
```

Appendix B: Dimension attributes

This appendix contains additional information about some of the dimensions and their attributes. An OLAP does not work in the same way as a relational database, for example, there is no schema or referential integrity.

However, some FAP dimensions have quite a few TM1 attributes. These attributes can be useful in various reporting scenarios as well as in TM1 scripts.

Company	
<language>_short	Short name in group language, <language> subject to set-up, for example, EN.
<language>_long	Long name in group language.
< language>_short	Short name in an additional language. Could be more than one additional (local) language.
< language>_long	Long name in an additional languages.
Structure Version	Period for the structure version, for example, 200911. Only on groups (aggregation nodes).
Consolidation Type	Consolidation type for the hierarchy, for example, CT LE. Only on groups (aggregation nodes).
Company Type	Company type as defined for the company in Controller.
Country Code	Country code as defined in Controller.
Local Currency	Local currency as defined in Controller.
Info 1 - Info 20	Info fields as defined in Controller (additional company information)

Account	
<language>_short	Short name in group language, <language> subject to set-up, for example, EN.

Account	
<language>_long	Long name in group language.
< language>_short	Short name in an additional language. Could be more than one additional (local) language.
< language>_long	Long name in an additional language.
Account Type	Account type as defined for the account in Controller.

Index