

# Understand Scenarios

Scenarios are private work areas in which users can model different assumptions within the Essbase data and see the effect on aggregated results, without affecting the existing data.

Each scenario is a virtual slice of a cube in which one or more users can model data and then commit or discard the changes.

Scenario-enabled cubes have a special dimension called Sandbox. The sandbox dimension is flat, with one member called Base and up to 1000 other members, commonly referred to as sandbox members. All members in the sandbox dimension are level-0.

Sandbox members are named sb0, sb1, and so on. Each sandbox is a separate work area, whereas the Base holds the data currently contained in the cube. A specific scenario is associated with exactly one sandbox member.

Sandbox

Base

sb0

sb1

sb2

Base data is the starting point before you use the sandbox to model possible changes.

Sandbox data (also known as scenario data) is not committed unless the scenario owner applies it, at which point it overwrites the Base data.

When first created, sandbox member intersections are all virtual and have no physical storage. The physical data from the cube is stored in the Base member slice. Querying new sandbox members dynamically reflects the values stored in the Base.

Only after you update any of the values in a sandbox are your changes stored physically in the sandbox. After you update some values in a sandbox member, queries against the sandbox reflect a mixture of stored sandbox values and values inherited dynamically from the Base.

Changes made in a sandbox are not committed to the Base until you do so explicitly, generally after an approval workflow. See [Understand Scenario User Roles and Workflow](#).

After you're finished with the sandbox, you can put the sandbox through the approval workflow, or you can skip the workflow and commit the updated values to the Base, or reject and discard the sandbox changes.

You must enable hybrid mode for scenario management to work. For queries, it is enabled by default. Do not disable it. For calculations, you also need to enable the [HYBRIDBSOINCALCSCRIPT](#) application configuration. See [HYBRIDBSOINCALCSCRIPT](#) (or use the [SET HYBRIDBSOINCALCSCRIPT](#) calculation command to control it on a per-calculation basis).

Security and filters apply to the Sandbox dimension.

Scenario enabled cubes have a CellProperties dimension that you should ignore, as it is for internal processes. You do not need to modify it nor account for it in calculations, queries, or load rules, and it shouldn't be included in any calculations or other operations.

## View and Work with Scenario Data

There are two entry points for viewing and working with scenario data in Smart View.

You can use the Essbase web interface to launch a scenario in Smart View, or you can use a Smart View private connection and work with the scenario data that way.

To analyze data in a scenario, you must have all of the following permissions:

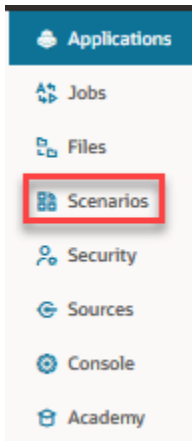
- Be a user provisioned to the application.
- Have a minimum of database access permission for the application (and have a write filter if you want to change data in the sandbox).
- Be a participant in the scenario (created by a user with higher privilege).


## View and Work With Scenario Data From the Essbase Web Interface

You can launch Smart View from a scenario in the web interface.

When you do this, because you enter from the scenario, you can only work in Smart View in the sandbox member associated with the scenario from which you entered. The sandbox member is implicit. You will not see it in the Smart View grid.

1. In the Essbase web interface, on the Home page, click **Scenarios**.



2. Click the Excel icon  next to the scenario you want to view.
3. Select to open the file.
4. This launches Excel with a Smart View connection to the scenario.

When you do this, the slice of data for that specific scenario is in the worksheet. You can query data only in that scenario. If you have minimum database update permission on the application, you can submit data to the scenario. (When you submit data to a scenario, you are submitting data to one sandbox member).

You can launch a scenario in Smart View from the web interface only on Windows using Firefox, Internet Explorer, or Chrome browsers.

## View and Work With Scenario Data From a Smart View Private Connection

You can open Excel and make a private connection to your cube, without starting from the web interface.

When you do this, the sandbox dimension will be in the worksheet, so you can submit data to any sandbox member to which you have access. This is helpful when you are a participant in more than one scenario, but you must explicitly know which sandbox you want to work in.

To see which sandbox member is associated with a scenario, go to the web interface, click on **Scenarios**, click on the scenario name, and view the **General Information** tab.

1. Open Excel.

2. Make a private connection to your scenario-enabled cube.
3. Do an ad hoc analysis.
4. Drill into the Sandbox dimension to view the sandbox members.

## Examples

This is a Smart View grid including the Base member and a sandbox member. Sandbox values have not been updated, so they reflect the Base values. Those values are stored only in the Base, not in the sandbox members:

					Base	sb10
Cola	New York	Actual	Jan	Sales	678	678
Cola	New York	Actual	Jan	COGS	271	271

The changed sandbox value below, 500, is stored in a sandbox member. The remaining sandbox value, 271 that was not updated is stored only in the Base:

					Base	sb10
Cola	New York	Actual	Jan	Sales	678	500
Cola	New York	Actual	Jan	COGS	271	271

Below is a grid with multiple sandbox members. If you have the Database Access user role and the appropriate write filter, you can submit data within multiple scenarios simultaneously:

					Base	sb0	sb1
Actual	Jan	Sales	New York	Cola	678	500	600
Actual	Jan	COGS	New York	Cola	271	271	271

## About Scenario Calculations

When you want to calculate a sandbox dimension in a scenario-enabled cube, you must include the sandbox members you want calculated in a FIX statement.

Otherwise, Essbase calculates only the base member.

By default, in non scenario-enabled BSO cubes, Essbase calculates all members from a dimension unless a FIX statement is used to limit the scope of the calculation to a specific member or group of members from the dimension.

The sandbox dimension is an exception to this behavior. If members from the sandbox dimension are not included in the FIX for a calculation, only the base member from the sandbox dimension is calculated by default. To calculate non-base members from the sandbox dimension, include them in the FIX statement, optionally along with the base member.

When you specify non-base sandbox members in a FIX statement, base is excluded from the calculation unless explicitly added into the FIX.

This behavior is different from calculations on non-sandbox dimensions excluded from the FIX; if you exclude a dimension from your FIX statement, Essbase calculates all members from the implied dimension. Sandbox dimensions are calculated differently, as the intent is usually to calculate either Base or specific sandboxes at a given time. Essbase calculates the Base member values, rather than the working sandbox values, except:

- When the calculation fixes on particular sandbox members.
- When the calculation is executed from a sheet launched from a scenario in the web interface (this is called a scenario-launched sheet). See [View and Work With Scenario Data From the Essbase Web Interface](#).
- When a sandbox cell value is selected in a private connection Smart View sheet and a calculation script is launched.

If you execute a calculation script from a scenario-launched sheet, the calculation runs in the sandbox associated with the scenario as long as no sandbox is explicitly mentioned in the script.

If you're in a sheet opened using a Smart View private connection and you're displaying sandbox and base values, if you highlight any data cell from the sandbox and launch a calculation script without explicit sandbox FIX, the sandbox will implicitly be calculated and Smart View will indicate that the sandbox was calculated. If you highlight a cell from the base member (or highlight no cell), then the base will be calculated when you launch your calc script and Smart View will indicate that the base was calculated.

You can calculate sandbox members using your pre-existing MaxL scripts by using the reserved runtime substitution variable name: `ess_sandbox_mbr`.

The following statement can be implemented (for your sandbox) in any MaxL script without creating any substitution variable on the server or application.

```
execute calculation appname.dbname.calcname with runtimesubvars  
'ess_sandbox_mbr=sb0';
```

## About Data Loads to Scenario-enabled Cubes

You can load scenario-enabled cubes using data exports taken before enabling the cube for scenarios. The data will load to the base sandbox member.

If you didn't use column export, then you can't have outline member changes that would invalidate your data load. If you used column export but your outline has changed, you may need a .rul file to load the data.

## About Data Exports from Scenario-enabled Cubes

Scenario-enabled Essbase cubes have a CellProperties dimension that is included in data exports, and must be considered when loading exported data. Also, it is important to understand the behavior of the sandbox dimension when working with exported data.

The following are considerations when exporting data from scenario-enabled cubes:

- If you use the web interface **Jobs** page to export data from a scenario-enabled cube, the resulting data file contains all three members from the CellProperties dimension (EssValue, EssStatus, and EssTID). Do not eliminate any of these columns.
- The data file from the export includes data physically stored in the cube, based on the selection you make: level zero data, all data, or input data.
- If values have been changed in sandboxes, then sandbox values will be in your export.
- In order to load exported data into sandboxes, values for all three CellProperties members (EssValue, EssStatus, and EssTID) must be in the data file.

## About Transparent and Replicated Partitions in Scenario-enabled Cubes

Transparent and Replicated partitions connect slices from two Essbase cubes together. This is the case when neither, one, or both cubes are scenario-enabled.

Sandboxes come into use when scenarios are created. However, there is no guarantee that scenarios on partitioned cubes will map to the same sandbox number. The same user may not be a participant in sandboxes in multiple cubes. Introducing scenarios imposes the following limitations:

- If source of a transparent partition is scenario-enabled, target queries will always pull data from the source base sandbox member.

- Write-back between scenario-enabled source and target cubes is only allowed between base members in the cubes, target cube base to source cube base.

Example: Write-back to source, which is normally enabled from transparent partition target cubes, is disabled for non-base sandbox members of scenario-enabled target cubes. It is a violation of permissions to allow a remote sandbox user to write directly into the base of the source cube.

- For replicated partitions, replication is only possible between source cube base and target cube base.

See [Understand Transparent and Replicated Partitions](#).

## About @XREF/@XWRITE in Scenario-enabled Cubes

In a scenario-enabled Essbase cube, you can use the calculation functions @XREF and @XWRITE to reference or write to data in another cube.

[@XREF](#) queries a remote cube from a local cube (the cube containing the @XREF statement). If the remote cube is scenario-enabled, @XREF only pulls base data from the remote cube.

[@XWRITE](#) updates a remote cube from a local cube (the cube containing the @XWRITE statement). Because @XWRITE writes data into the remote cube, the scope of the @XWRITE statement matters.

For different combinations of scenario-enabled and non-scenario-enabled cubes, @XWRITE behaves in the following ways:

When a scenario-enabled local cube references a non-scenario-enabled remote cube,

- A [FIX](#) on the base member in the local cube with an @XWRITE to the remote cube writes the local cube base into the remote cube.
- No FIX on any sandbox member in the local cube with @XWRITE to the remote cube writes the local cube base into the remote cube. If you don't include a sandbox member in the Fix, base is included automatically.
- A FIX on sandbox in the local cube with an @XWRITE to the remote cube returns an error. Writing from a non-base sandbox member into a remote cube is not supported.

When a scenario-enabled local cube references a scenario-enabled remote cube,

- A FIX on the base member in the local cube with an @XWRITE to the remote cube writes the local cube base into the remote cube base.
- No FIX on any sandbox member in the local cube with an @XWRITE to the remote cube writes the local cube base into the remote cube base. If you don't include a sandbox member, base is included automatically.
- A FIX on sandbox in the local cube with an @XWRITE to the remote cube returns an error. Writing from a non-base sandbox member into a remote cube is not supported.

When a non-scenario-enabled local cube references a scenario-enabled remote cube, @XWRITE always updates the remote cube base member.

See [Understand @XREF/@XWRITE](#).

## About Audit Trail in Scenario-enabled Cubes

Data audit trail tracks updates made to data in a cube. To work with audit trail in scenario-enabled cubes you, should understand what defines "old" and "new" data values, and the two different entry points for working with sandbox data in Smart View.

This topic assumes you are familiar with the different entry points for viewing scenario data. See:

- [View and Work With Scenario Data From a Smart View Private Connection](#)
- [View and Work With Scenario Data From the Essbase Web Interface](#)

If you consider the latest data update committed to a cell to be “new” data, and all prior data values for that cell to be “old,” it can help you understand how audit trail works in scenario-enabled cubes.

A new or unused sandbox in a scenario-enabled cube contains no stored values. The values shown to users, such as the values displayed in a spreadsheet, reflect the values stored in the base.

If you use data audit trail on a new scenario-enabled cube, the base values that display in the spreadsheet for the sandbox are considered the “old” values.

When you update values in a sandbox, those values are stored in the sandbox (not in the base). For the purposes of data audit trail, these values are the “new” values.

If you later update these “new” values, audit trail will track the latest changes. It will treat the previous values as “old” and the updated values as “new.”



In summary,

- Old values are the base values reflected in a new sandbox.
- Initially, new values are the updated, stored values in the sandbox.
- Subsequently, updated values are new, and the values they replace are old.

There are two possible entry points for working with data in Smart View:

- Where you open Excel and make a private connection to your cube, without starting from the Essbase web interface.
- Where you launch Smart View from a scenario in the web interface.

When you start by opening Excel and making a private connection to your cube, audit trail works as you would expect with any other data set.

When you launch Smart View from a scenario in the Essbase web interface, audit trail works differently.

- When you export logs to a sheet, the sheet does not show the implicit sandbox member.
- When you launch a new sheet using the **Ad hoc** button below the **Audit Trail** pane, the new sheet does not show the implicit sandbox member, and any changes in that sheet affect the data values for that sandbox member.

## About Scenario Limitations

These limitations apply to scenarios and sandbox dimensions.

- Scenarios are not supported on aggregate storage cubes.
- The DATAEXPORT calculation command isn't supported on sandbox members. It is only supported on the Base member.
- When you connect to a scenario from a scenario-launched sheet, MDX queries, MDX inserts, and MDX exports will work with the base instead of working with the sandbox for that scenario.
- Runtime substitution variables with the svLaunch parameter are not supported when you launch the scenario in Smart View from the Essbase web interface.  
See [View and Work With Scenario Data From the Essbase Web Interface](#).

Runtime substitution variables with the svLaunch parameter work correctly when you connect to the scenario directly from a private connection. This is because the sandbox member is included in the sheet.

There are a limited number of functions that are not supported in hybrid mode, which is used with scenario-enabled cubes. See [Functions Supported in Hybrid Mode](#).