



**HYPERION® SYSTEM™ 9 PERFORMANCE
SCORECARD™ IMPORT \ EXPORT
FACILITY**

RELEASE 9.2.0.3

ADDENDUM



CONTENTS IN BRIEF

The Import/Export Facility	2
Requirements	2
How the Facility Works	2
Installing the Facility	14
Importing and Exporting Data Using Behaviours	14

The Import/Export Facility

The Import/Export Facility is a standalone application that writes directly to the Performance Scorecard database. It enables you to share data between Performance Scorecard and comma-separated value (CSV) text files by performing these tasks:

- Exporting data from a Performance Scorecard application to a CSV file
- Importing data from a CSV file to Performance Scorecard applications

You can combine these operations to extract data from Performance Scorecard, modify then reimport the modified data in one operation using a scheduling program.

You can also use Microsoft Excel spreadsheets to create a data file which can update, add, or delete data in Performance Scorecard, or export existing data from Performance Scorecard for use in another environment. This also enables you to extract all computed values such as targets, results, scores and status and trend information.

TIP: Imported data is incorporated in Performance Scorecard applications after 150 seconds. You need not restart the application server.

Invoke the facility from the command line using an import or export command. Each command uses the same line syntax. Specify the behavior to use and a data file in the command. The behavior defines the type of data to import or export. The data file is the name of the file that contains or will contain the data.

Requirements

These elements are required to import or export data:

- `Config.properties` — Contains the required database connection information, date, number formats, and so on. Most of these settings can also be specified as command line options. `Config.properties` — Customizable file created during installation. .
- CSV file that contains the data to import or export. The format differs for each behavior.
- The behaviors used to import or export different business objects. Behaviors have short names to simplify the commands you will run. For example, `msr` for Measure.

How the Facility Works

This section describes how the following elements are used to import and export data:

- [“CSV Files” on page 3](#)
- [“Action” on page 3](#)
- [“Behaviors” on page 5](#)
- [“Configuration File” on page 6](#)
- [“Sample Configuration File” on page 8](#)
- [“Logging Errors” on page 9](#)

CSV Files

The CSV file uses Microsoft Excel formatting conventions and must contain a header row that defines the columns used and the order in which they are displayed. Although most columns are optional, the Name and Action columns are mandatory. Each row, other than the header row, defines a record to be input.

See:

- [“Action” on page 3](#)
- [“Name or Key” on page 3](#)
- [“Data Rows” on page 4](#)

You can place columns in any order. When importing, the facility recognizes the name of the column in the header. When exporting, all possible columns for the behavior are exported. Use a query file to specify the columns to export, and the order in which to export them. See [“Using Queries” on page 12](#).

Action

The Action column is always required.

Table 1 Action Values

Value	Description	Action on Import	Action on Export
V	View	Ignore the record	The action column is automatically set to “V” when exporting
C	Create	Create a new object as defined by the row.	Not valid for export.
U	Update	Update all fields for the object specified by the name or key. Any previous values are overwritten.	Not valid for export.
D	Delete	Delete the record specified by the name or key.	Not valid for export.

Because “V” records are ignored on import, it is faster, when performing updates of a few records, to export all the data, and then modify the records to be changed. Change the action values of the records to be changed to “U” or “D,” and then import the entire file. The “V” records are ignored, and only those records set to “U” or “D” have any effect.

Name or Key

Because objects are uniquely identified by name, the Name column identifies the object to update or delete. Note that, for employees, the E-mail column is the unique identifier, since names are likely to be duplicated.

In cases where multiple objects have the same name, the primary key of the object can be used as the identifier in the key column. If present, the key column takes precedence over the name column.

Note:

When matching names, the match is not case-sensitive. Any leading or trailing spaces are stripped before matching.

Data Rows

A data row must contain a column for every column in the header, in the same order. These types of data may be contained in a single column:

Table 2 Data Types

Data Type	Description
String	A generic string of characters, which may contain any characters.
Integer	An integer, which may contain numbers, plus (+) or minus (-) signs.
Decimal	A decimal number that is formatted as defined in the configuration file.
Date	A date that is formatted as defined in the configuration file.
Formula (String)	A string representing a formula. The formula must follow the syntax defined for Performance Scorecard formulas. See the <i>Hyperion System 9 Performance Scorecard Application Designer's Guide</i> .
0-1 (type)	These values represent a relationship to a maximum of one other object. The value shown in brackets is the type or list of types supported. For example, 0-1 (Employee). The object may be specified by name, or primary key. If it is a primary key, the value must be delimited within \${}. For example \${32f.-1.-f419e0}.
1-1 (type)	These values represent a mandatory relationship to one other object. The value shown in brackets is the type or list of types supported. For example, 0-1 (Employee). The object may be specified by name, or primary key. If it is a primary key, the value must be delimited within \${}. For example \${32f.-1.-f419e0}.
0-n (type)	These values represent a relationship to one or more other objects. The value shown in brackets is the type or list of types supported. For example, 0-n (Employee).
Constant	Predefined constant value must be used. Although the case of the constant name is not important, the name must match exactly. Available values for the type are included in the field description.

You must follow the CVS file formatting rules. If you use Excel or other software to generate the file, this is done automatically.

If you use a text editor, follow the formatting rules to avoid errors processing the data. For example, if the data in a column contains a comma, the column must be enclosed in quotes. Refer to documentation on standard CSV format for detailed rules.

Behaviors

When importing or exporting data, you must specify the behavior or associated object that is to be performed. The behavior defines the file format. Behavior names are not case-sensitive.

You can use the short form of the behavior in the command line prompts, such as `msr` for Measure, or `scd` for Scorecard.

Table 3 Import and Export Behaviors

Behavior	Short Form	Description
AccountabilityElement	ame	Import or export Accountability map elements (AME)
AccountabilityMap	amap	Import or export Accountability maps
Dimension	dim	Import or export custom dimensions
DimensionMember	dm	Import or export custom dimension members
Employee	emp	Import or export employee records
GroupAccount	group	Import or export group accounts
Initiative	ini	Import or export initiatives
Measure	msr	Import or export measure
MeasureFactory	mfc	Import or export measure templates
MeasureData	mdata	Import or export measure results, scores, and targets
MeasureTargetFormula	mtf	Import or export the formulas that compute measure targets
Scorecard	scd	Import or export scorecards
ScorecardElements	sce	Import or export scorecard elements
StrategyMap	smap	Import or export Strategy maps
StrategyMapElement	sme	Import or export strategy elements (SME) used on Strategy maps
Target	tar	Import or export target definition, not target values
Unit	uni	Import or export units
UserAccount	user	Import or export user accounts
Variable	var	Import or export variables
VariableResults	vrs	Import or export variable results

Configuration File

`Config.properties` defines the database connection required to import and export data. By default, the file is in: `c:\Hyperion\hps\9.2.0.3\tools\config\config.properties`, or in `<HYPERION_HOME>`.

Although you can override default settings in the file from a command line, make permanent changes in the file. For example, if your data uses a different date format, modify that setting in the file.

The following table lists the entries in `config.properties` that you may want to change. You must adhere to the date and decimal syntax in [Table 5](#) and [Table 6](#).

Table 4 Customizing `Config.properties`

Entry	Description
<code>hpsconfig</code>	Location of a Performance Scorecard configuration file, such as <code>HPSCConfig.properties</code> . This file contains the database connection and other information required to start a connection to the Performance Scorecard repository.
<code>dateFormat</code>	Data format to use for parsing or formatting dates. The formatting codes are also those that set the date format in Performance Scorecard user preferences. See Table 5
<code>numberFormat</code>	Format for decimal numbers. See Table 6 .
<code>codePage</code>	Code page of the data file. If there is a query file, it must also be in this code page.
<code>errorLog</code>	Location of the error details log.
<code>failedRecordsLog</code>	Location of the log where failed record events are written.
<code>successfulRecordsLog</code>	Location of the log where successful record events are written.
<code>outputFile</code>	Location of the file where records that failed to import are written.
<code>queryFile</code>	On export, location of the file specifying the records to be extracted.
<code>mappingFile</code>	Location of the mapping file. This file (defined below) specifies the mapping between record fields and object attributes in the application.
<code>username</code>	The username to authenticate with when connecting to the repository.
<code>password</code>	The password to authenticate with when connecting to the repository.
<code>domain</code>	Optional: Used only for NTLM authentication.
<code>nolock</code>	If specified, no checks will be made to see if someone is editing an object before updating or removing it. Do not specify this if there is any possibility anyone will be editing data in Performance Scorecard while the import/export is in progress.
<code>locale</code>	The locale to use. This affects a few things: it will specify the default number and date format if none is specified in this file; it also specifies the language to use for help messages and event log entries.

Entry	Description
rowTransactions	<p>If set to true (default), each record is treated as a database transaction. An error in one record will not affect the processing of other records.</p> <p>If set to false, the entire import is treated as one transaction. This setting improves performance; however, any error will cause the entire job to fail .</p>
module.driver	Specifies class of the driver module.
module.parser	Specified class of the Parser module.
module.resolver	Specifies class of the resolver module
module.repository_manager	Specifies class of the repository module

Table 5 Date Formats

Format	Component	Presentation	Example
G	Era designator	Text	AD
y	Year	Year	2006; 06
M	Month in year	Month	July; Jul; 07
w	Week in year	Number	27
W	Week in month	Number	2
D	Day in year	Number	189
d	Day in month	Number	10
F	Day of week in month	Number	2
E	Day in week	Text	Tuesday; Tue
a	AM/PM marker	Text	PM
H	Hour in day (0-23)	Number	0
k	Hour in day (1-24)	Number	24
K	Hour in am/pm (0-11)	Number	0
h	Hour in am/pm (1-12)	Number	12
m	Minute in hour	Number	30
s	Second in minute	Number	55
S	Millisecond	Number	978
z	Time zone	General Time Zone	Pacific Standard Time; PST; GMT-08:00
Z	Time zone	RFC 822 time zone	-0800

Table 6 Decimal Format

Symbol	Location	Localized	Description
0	Number	Yes	Digit
#	Number	Yes	Digit, zero shows as absent
.	Number	Yes	Decimal separator or monetary decimal separator
-	Number	Yes	Minus sign
,	Number	Yes	Grouping separator
E	Number	Yes	Separates mantissa and exponent in scientific notation. This does not need to be quoted in prefix or suffix.
;	Subpattern boundary	Yes	Separates positive and negative subpatterns.
%	Prefix or Suffix	Yes	Multiply by 100 and show as percentage.
\u2030	Prefix or Suffix	Yes	Multiply by 1000 and show as per mille.
¤ (\u00A4)	Prefix or Suffix	No	Currency sign, replaced by currency symbol. If doubled, it is replaced by the international currency symbol. If present in a pattern, the monetary decimal separator is used instead of the decimal separator.
'	Prefix or Suffix	No	Used to quote special characters in a prefix or suffix, for example, "'#' formats 123 to "#123". To create a single quote, use two in a row: "# o'clock".

Sample Configuration File

```

hpsconfig=config\HPSConfig.properties
dateFormat=dd/MMM/yyyy
numberFormat=#.##
codePage=unicode
errorLog=log\error.log
failedRecordsLog=log\failedRecords.log
successfulRecordsLog=log\successfulRecords.log
#outputFile=
#queryFile=data\query.csv
mappingFile=config\EtlMapping.xml
username=designer
password=password
domain=
nolock=false
locale=en
rowTransactions=true # Stretch to support whole file as 1 transaction,
for performance

# Module definitions
module.driver=com.hyperion.pmd.hps.etl.parser.CSVDriver
module.parser=com.hyperion.pmd.hps.etl.parser.ParserModule
module.resolver=com.hyperion.pmd.hps.etl.handler.ScorecardResolverModule

```

```
module.repository_manager=com.hyperion.pmd.hps.etl.handler.ScorecardRepositoryManagerModule
```

Logging Errors

By default, these log files record the progress of data transfer and display any errors:

- `errors.log` — Describes any errors that occur. You can configure this file to display warning and debug messages.
- `failedRecords.log` — Contains an entry for each record that fails to load, noting the record number, and, if possible, the name of the object the record represents.
- `successfulRecords.log` — Contains an entry for each loaded record. Because it will be large, you must modify the configuration file to generate this file.

To modify the Configuration file setting or to use `successfulRecords.log`, see [“Optional Command Line Arguments” on page 9](#).

Optional Command Line Arguments

In addition to the standard parameters required for the import or export commands, there are also a number of optional arguments available. These commands are appended to the end of the standard import or export command. You can add multiple optional commands.

Table 7 Optional Command Line Arguments

Command	Description
<code>-c <configfile></code>	Specify the path to the configuration file.
<code>-l <logfile></code>	Specifies the path to the log file to which any unexpected errors are written. If another file is not specified, the default log file is <code>errors.log</code> .
<code>-e <failedRecordsLogs></code>	Specifies the path to the log file for any failed records. Specially formatted messages describes events that prevented records from being read or written. If not specified, the default file is <code>failedRecords.log</code> .
<code>-i <successfulRecordsLog></code>	Specifies the path to the log file for all successful records that were read or written to. By default, this log is not written because of its size. If required, you can set this log in the configuration file.
<code>-o <outputfile></code>	For import, specifies the path to the output file where error records are written. By default, this is the name of the file, with “output” added, in the data directory. For example, the input file <code>employees.csv</code> would have an output file of <code>employees.output.csv</code> .
<code>-q <queryfile></code>	For export, specifies a file containing the query information to determine which records are to be exported. If not specified, all records are exported.
<code>-m <mapping></code>	Specifies the path to the mapping file, that defines the format of the expected CSV file.

Command	Description
	The configuration file points to this file but may be overridden by this option. The mapping file only requires information about a single behavior, so may prove to be useful if your input files are in a slightly different but compatible format.
-d <date format>	Specifies the date format. This option overrides the configuration file setting.
-n <number format>	Specifies the number format. This option overrides the configuration file setting.
-cp <codepage>	Specifies the code page to use when reading or writing the CSV file. This option overrides the configuration file setting.
-u <username>	Specifies the Performance Scorecard user name to log on to the application. This option overrides the configuration file setting.
-p <password>	Specifies the Performance Scorecard password to log on to the application. This option overrides the configuration file setting.
-domain <domain>	Specifies the NTLM domain, if you are authenticating with NTLM.
-nolock	Specifies that no attempt should be made to obtain business object locks before updating or deleting. Caution! Use this option only if there are no other current users on the system.
-v <config file option>	Define any configuration file or custom option. For example: -v rowTransactions=false is equivalent to setting rowTransactions=false in the configuration file.

Importing Data

You can import data from a comma-separated value (CSV) text file into Performance Scorecard using the Import/Export Facility.

➤ To import data:

- 1 In Microsoft Excel, or another application, create the CSV file that contains the data to be imported into Performance Scorecard.

To ensure a successful import, the CSV file must use the appropriate file formats.

- 2 Open a command window and change to the directory containing the import and export commands. The default location on Windows is C:\Hyperion\hps\9.2.0.3\tools\bin, or in your custom <HYPERION_HOME> directory.

- 3 Enter the import command, using this format:

```
import <behavior> <filename.csv> [options]
```

For example:

```
import emp employee.csv
```

When the processing is complete, a message is displayed.

- 4 Launch Performance Scorecard to verify the changes.

Exporting Data

You can export data from Performance Scorecard to a comma-separated value (CSV) text file. You can export all data for an object type, such as all employees, or you can limit the number or type of objects and dates for export.

► To export data:

- 1 Open a command window and change to the directory containing the import and export commands. The default location on Windows is `C:\Hyperion\hps\9.2.0.3\tools\bin`.

- 2 Enter the export command, using this format:

```
export <behavior> <filename.csv> [options]
```

For example:

```
export emp employee.csv
```

When the processing is complete, a message is displayed.

- 3 In Microsoft Excel, or other application, open the file to view the export file.

All the information you requested in the command is in the file.

- 4 Update, edit or delete data in the file.

- 5 Save the file.

Updating and Modifying Data

CSV files enable you to export data from Performance Scorecard, modify or delete the data, and re-import it into Scorecard in the same session.

► To perform a round trip data transfer:

- 1 Open a command window and change to the directory that contains the import and export commands. By default, this is `C:\Hyperion\hps\9.2.0.3\tools\bin`.

- 2 Enter the export command, using this format:

```
export <behavior> <filename.csv> [options]
```

- 3 In Microsoft Excel, or other text application, open the CSV file to view the export file.

- 4 Edit the data in the CSV file to add, remove or modify any entries. For each row, set the action required to reflect the change:

- V — Set to V (View) if no change was made to extract the record, but ignore the row when re-importing data.
- C — Set to C (Create) if you added a new object, as defined by the row.

- **U** — Set to U (Update) if any data was modified. Changes are picked up when re-importing data.
- **D** — Set to D (Delete) to delete a record specified by name or key.

You must, at a minimum, change the name to something unique. If you are creating objects, remove the Key and ID columns, as these must be automatically generated by the system. As a general guideline, define your create records in a separate file.

Caution!

If any columns are left blank, they will overwrite any data in the original database.

- 5 **Save the CSV file.**
- 6 **In the Run dialog box in **Open**, type `cmd`, then click **OK**.**
The Command window is displayed.
- 7 **On the command line, enter `cd <path to Import.bat>`.**
- 8 **Enter the import command, using this format:**

```
import <behavior> <filename.csv> [options]
```

Any changes, additions or deletions are re-imported into Performance Scorecard. When the processing is complete, a message is displayed.

- 9 **Launch Performance Scorecard to verify the changes.**

Using Queries

To limit the data extracted during an export, create a simple query. A query file follows the same behavior you are exporting. Only columns specified in the header of the query file are extracted, in the order specified in the query file. The query file can contain multiple rows, and each row specifies a record to extract.

To identify the objects to extract, specify either names or keys. Wild cards are not supported, so you must specify one record, with the exact name, for each object to extract. In the case of measure data, you must also specify a date for the data to be extracted.

These query types are supported:

- Name/Key query
- Date query

➤ To run a Name/Key query:

- 1 **In Microsoft Excel, or another text application, create a CSV text file that contains the Name, Key or ID column of the object to be exported.**
- 2 **Optional: Add a new row for each additional Name in the query.**

Note:

Wildcard characters are not supported in a query.

- 3 Save the CSV file.

- 4 From the Windows main menu, select **Start > Run**.

The Run dialog box is displayed.

- 5 In **Open**, type `cmd`, then click **OK**.

The Command window is displayed.

- 6 On the command line, enter `cd <path to Import.bat>`.

By default, the `.bat` or `.sh` file is installed in `C:\Hyperion\hps\9.2.0.3\Tools\bin`.

- 7 Enter the import command, using this format:

```
export <behavior> <filename.csv> -q <query.csv
```

When the processing is complete, a message is displayed.

- 8 In Excel, open the exported data file for viewing.

➤ To run a Date query:

- 1 In Microsoft Excel, or another text application, create a CSV text file that contains the Action and Date for which you want data to be exported.

For example:

Table 8 Date Query Example

Action	Date
V	01/01/2006

- 2 **Optional:** Add a new row for each additional date in the query.

Note:

To extract results for all objects, leave the name and key columns blank, or do not include them at all.

- 3 Save the file.

- 4 From Windows, select **Start > Run**.

The Run dialog box is displayed.

- 5 In **Open**, type `cmd`, then click **OK**.

The Command window is displayed.

- 6 On the command line, enter `cd <path to Import.bat>`.

By default, the `.bat` or `.sh` file is installed in `C:\Hyperion\hps\9.2.0.3\Tools\bin`.

- 7 Enter the import command, using this format:

```
import <behavior> <query.csv> [options]
```

When the processing is complete, a message is displayed.

- 8 In Excel, open the CSV file to view the query file.
- 9 Save the changes in the CSV file.

Installing the Facility

This section describes the two ways to install and use the facility.

Installing on the Same Computer as Performance Scorecard

The facility is automatically installed with this 9.2.0.3 release of Performance Scorecard. The database for the facility is the same as the one that you configure for Performance Scorecard. The facility installs to: *<Performance Scorecard installation>*\9.2\tools\bin. You will use these files:

- Microsoft Windows: `export.bat` and `import.bat`
- UNIX: `export.sh` and `import.sh`

Installing Just the Facility on Another Computer

To install only the facility on a second computer, perform these tasks:

1. Install and configure Performance Scorecard on one computer.
2. Install Performance Scorecard on another computer. Configure this instance as follows:
 - Reference the database used with the first instance of Performance Scorecard. Both instances must use the same database.
 - Deploy to the application server. This need not be the same application server you used to deploy the first instance of Performance Scorecard.

Importing and Exporting Data Using Behaviours

For each behavior, you must use the appropriate file format to ensure the successful transfer of data.

In the file formats for each behavior, “R.” indicates required data and “K” identifies business objects, such as measures.

See:

- [“AccountabilityElement” on page 15](#)
- [“AccountabilityMap” on page 16](#)
- [“Dimension” on page 16](#)
- [“DimensionMember” on page 17](#)

- “Employee” on page 17
- “GroupAccount” on page 18
- “Initiative” on page 19
- “Measure” on page 20
- “MeasureTargetFormula” on page 25
- “Measure Factory” on page 24
- “MeasureTargetFormula” on page 25
- “Scorecard” on page 25
- “ScorecardElements” on page 27
- “StrategyMap” on page 27
- “Target” on page 29
- “Unit” on page 30
- “UserAccount” on page 31
- “Variables” on page 31
- “VariableResult” on page 32

AccountabilityElement

Use this behaviour to import or export accountability elements.

Table 9 AccountabilityElement

Required Fields and Key	Name	Description	Type	Size
R	Action	C, V, U, D	Constant	
K	Name	Element Name	String	255
	Key	Primary Key	String	
	ID	Object ID	Integer	
	Description		String	1000
R	Map Name	Name of the map on which the element is located.	1-1 (Accountability Map)	
R	Parent	Name of the parent map element. If blank, the element is on the root level of the map.	0-1 (SME)	
R	Order	Ordering of the map element, relative to other map elements that are children of this element.	Integer	

Required Fields and Key	Name	Description	Type	Size
	Label		String	255
	Title	Title for map element. This value overrides the default element name.	String	255
	Mission		String	1000
	Vision		String	1000
	Differentiators		String	1000
	Owners	Name of employees who are owners for this map element.	0- n (Employee)	
	Members	Name of any employees who are members for this map element.	0- n (Employee)	
	Annotation Creators	Name of any employees who are specified as annotation creators for this element.	0- n (Employee)	
	Strategy Elements	Name of any strategy elements associated with this element.	0-n (SME)	

AccountabilityMap

Use this behaviour to import or export Accountability maps.

Table 10 AccountabilityMap

Required Fields and Key	Name	Description	Type	Size
R	Action	C, V, U, D	Constant	
K	Name	Map Name	String	255
	Key	Primary Key	String	
	ID	Object ID	Integer	
	Description		String	2000
	Domains		0 – n (Domain)	

Dimension

Use this behaviour to import or export custom dimensions.

Table 11 Dimension

Required Fields and Key	Name	Description	Type	Size
R	Action	C, V, U, D	Constant	
K	Name	Name of the custom dimension This name must be unique across all dimensions.	String	255
	Key	Primary Key	String	
	ID	Object ID	Integer	
	Description		String	2000
	Order	Sequential position in the dimension list	Integer	

DimensionMember

Use this behaviour to import or export dimensional members.

Table 12 DimensionMember

Required Fields and Key	Name	Description	Type	Size
R	Action	C, V, U, D	Constant	
K	Name	Name of the dimension member. This name must be unique across all dimensions.	String	255
	Key	Primary Key	String	
	ID	Object ID	Integer	
	Dimension		1-1 (Dimension)	
	Parent	Parent of the dimension member. A parent is mandatory. The only dimension member that does not have a parent is the member that is automatically created with the dimension.	1-1 (Dimension Member)	
	Description	Dimension to which this member belongs	String	2000
	Order	Sequential position in the dimension list	Integer	

Employee

Use this behaviour to import or export employees.

Table 13 Employee Metadata

Required Fields and Key	Name	Description	Type	Size
R	Action	C, V, U, D	Constant	
K	Email	Email address of the employee. This address must be unique across all employees.	String	
	Key	Primary Key	String	
	ID	Object ID	Integer	
	First Name	The first name of the employee.	String	255
	Last Name	The last name of the employee.	String	255
	Employee Number	The internal employee record number, if available.	String	255
	Title	The job title for the employee	String	255
	Department	The name of the department or business unit	String	255
	Manager	Name of the manager for the employee	0-1 (Employee)	
	Domain	Name of the domain with which the employee is associated	0-n (Domain)	
	Business Address	Business address or office of the employee	String	255
	Home Address	Home address of the employee	String	255
	Home phone	Home telephone number for the employee	String	255
	Work phone	Business telephone number for the employee	String	255
	Cell phone	Cellular or mobile telephone number for the employee	String	255
	Userid	User Account ID associated with the employee	0-n (User Account)	

GroupAccount

Table 14 GroupAccount

Required Fields and Key	Name	Description	Type	Size
R	Action	C, V, U, D	Constant	

Required Fields and Key	Name	Description	Type	Size
K	Name	Name of the group account	String	
	Key	Primary Key	String	
	ID	Object ID	Integer	
	Description	Description of the purpose of the group account	String	200
	Roles	Security roles associated with the group	0-n (Role)	
	Parents	Groups of which this group account is a child	0-n (Group Account)	

Initiative

Use this behaviour to import or export initiatives.

Table 15 Initiative Metadata

Required Fields and Key	Name	Description	Type	Size
R	Action	C, V	Constant	
K	Name	Name of the initiative	String	255
K	Link Name	Name of object to which the initiative is attached. For import, an initiative is only associated with one owning object. For export, you can access the initiative through any of its associated objects.	String	255
K	Link Type	Type of object to which the initiative can be attached: <ul style="list-style-type: none"> ● Accountability Element ● Strategy Element ● Measure 	Constant	
	Description	Description of the initiative	String	2000
	Cost	The cost to complete the initiative	Decimal	
	Priority	How important the initiative is, from a customer-created set	0-1 (<i>Initiative Priority</i>)	
	Completion Date	Date that the initiative is actually completed	Date	

Required Fields and Key	Name	Description	Type	Size
	Cost Unit	Selected unit of the cost value, such as a currency	0-1 (Unit)	
	Effort	Effort required to complete the initiative, such as working hour		
	Notify	Name of the employee to be notified about the state of the initiative	0-n(Employees)	
	Owners	Name of the employees who are responsible for the initiative	0-n (Employees)	
	Start Date	Date on which the initiative begins	Date	
	Due Date	Date by which the initiative should be completed	Date	
	Status (Name)	The level of progress toward completing the initiative. In text.	String	
	Status (Symbol)	An icon or image that represents the level of progress toward completing the initiative.	Status (Name)	
	Percent Complete	The percentage of the initiative that is complete	Decimal	

Measure

Use this behaviour to import or export measures.

Table 16 Measure Metadata

Required Fields and Key	Name	Description	Type	Size
R	Action	C, V, U, D	Constant	
K	Name	Name of the measure For dimensional measures, this is the template name only. Dimensions are specified in the Dimensions column.	String	255
	Key	Primary Key	String	
	ID	Object ID	Integer	
	Description	Description of the Measure	String	2000

Required Fields and Key	Name	Description	Type	Size
	Dimension Members	<p>List of relationships to dimension members</p> <p>If blank, the measure is treated as a non-dimensional measure.</p> <p>If the measure is a dimensional measure template, the measure is automatically linked to that template, and the set of specified dimensions.</p> <p>Note: To avoid potential errors or confusion, create all dimensional measures first with a single file, then perform overrides, if any, with a second file.</p> <p>Caution! If you do combine them into a single file, you must specify <code>ignore</code> for any columns for which you do not want to override the value for the dimensional measure.</p>	0-n (Dimension Members)	
	Unit	Name of the unit	0-1 (Unit)	
	Result Frequency	<p>The frequency code for the result frequency:</p> <ul style="list-style-type: none"> ● None ● Daily ● DailyMF ● Weekly ● Semi-Monthly ● Monthly ● Quarterly ● Semi-Annually ● Annually 		
	Result Subfrequency	<p>The frequency code for the result subfrequency:</p> <ul style="list-style-type: none"> ● FIRST ● LAST ● SUNDAY ● MONDAY ● TUESDAY ● WEDNESDAY ● THURSDAY ● FRIDAY 		

Required Fields and Key	Name	Description	Type	Size
		<ul style="list-style-type: none"> ● SATURDAY ● DAY_OFFSET ● FIRSTSUNDAY ● FIRSTMONDAY ● FIRTTUESDAY ● FIRSTWEDNESDAY ● FIRSTTHURSDAY ● FIRSTFRIDAY ● FIRSTSATURDAY ● FIRST_DAY_OFFSET ● LASTSUNDAY ● LASTMONDAY ● LASTTUESDAY ● LASTWEDNESDAY ● LASTTHURSDAY ● LASTFRIDAY ● LASTSATURDAY ● LAST_DAY_OFFSET ● USER_DEFINED ● NO_WEEKEND 		
	Result Interval	<p>Number of periods between result points.</p> <p>For example, a setting of Monthly, with an interval of 2, results in a frequency of every other month.</p>	Integer	
	Result Day	<p>Number of days offset into the period.</p> <p>For example, Monthly with a result day of 17, is the 17th of every month</p>		
	Result Month	<p>For Annually frequency, month of period that contains the frequency.</p>	Integer	
	Collection Frequency	Result collection frequency	Constant	
	Collection Subfrequency	Result collection sub-frequency	Constant	
	Collection Interval	Result collection interval	Integer	
	Collection Day	Result collection day	Integer	
	Collection Month	Result collection month	Integer	

Required Fields and Key	Name	Description	Type	Size
	Extension	Collection extension (in days)	Integer	
	Start Date		Date	
	PTD Frequency	Frequency for period-to-date results This value must be a lower frequency than the Result Frequency.	Constant. See Result Frequency for values.	
	PTD Function	Function to be used when computing period-to-date values: <ul style="list-style-type: none"> ● sum ● avg ● min ● max ● none 	Constant	
	PTD Formula	Formula that calculates period-to-date values	Formula	
	Trend Offset	Number of periods to look back when calculating trend values. This value must be greater than or equal to 1.	Integer	
	Higher is Better	Indicates whether a higher result values represent better performance. Set to T (true) or F (false).	Constant	
	Result Formula		Formula	
	Score Formula		Formula	
	<target> Formula	The formula specified by <target>	Formula	
	Status Base	Type of status base: <ul style="list-style-type: none"> ● Result ● Score ● PTD 	Constant	
	Decimal Places	Setting for number of decimal places	Integer	
	Report Name	Name of the report	String	40
	Report URL	URL for the report	String	255
	Data Source	Name of the data source	String	40

Required Fields and Key	Name	Description	Type	Size
	New Window	Indicates whether the report will launch in a new Window. Set to T (true) or F (false).	Constant	
	Rollup Formula	Used for dimensional measures only.	Formula	
	Priority	Set to these values: <ul style="list-style-type: none"> ● 100 indicates primary measure ● 0 indicates a secondary measure 	Constant	
	Perspective	Primary perspective	0-1 (Perspective)	
	Secondary Perspective	Secondary perspective	0-n (Perspective)	
	Domains	Specified domain that contains the measure	0-n (Domain)	
	Owners	Employee who is the owner of the measure	0-n (Employee)	
	Result Collectors	Employee who is result collector for the measure	0-n (Employee)	
	Target Setters	Employee who is the target setter for the measure	0-n (Employee)	
	Annotation Creators	Employee who is an annotation creator for the measure	0-n (Employee)	

Measure Factory

Use this behaviour to import or export measure templates (factories).

Table 17 MeasureFactory

Required Fields and Key	Name	Description	Type	Size
For Measure Templates, use the same file formats specified in “Measure” on page 20 and add this behavior.				
	Dimension Members	Top level dimension members to be associated with the template	0-n (Dimension Member)	

Measure Data

Use this behavior to import or export measure results, targets, and scores.

Table 18 MeasureData

Required Fields and Key	Name	Description	Type	Size
R	Action	C, V, U, D	Constant	
K	Name	Name of the measure	String	255
	Key	Primary Key	String	
	ID	Object ID	Integer	
K	Date	Date for which results were generated	Date	
	Result	Result value	Decimal	
	Target <target>	Target value for the specific target	Decimal	
	Score <target>	Read only. Score calculated, using the specific target.	Decimal	
	PTD	Read only. Period-to-date value	Decimal	
	Status <target>	Read only. Name of status, calculated, using the specific target.	String	255
	Trend	Indicates trend: <ul style="list-style-type: none"> ● TUG – trend is up, higher is better ● TDG – Trend is down, and higher is worse ● TUP – Trend is up, and higher is worse ● TDP – Trend is down, and higher is better ● TS – Result unchanged from last period 	Constant	

MeasureTargetFormula

Use this behaviour to import or export formulas that compute measure targets.

Table 19 MeasureTarget Formula

Required Fields and Key	Name	Description	Type	Size
R	Action	C, V, U, D	Constant	
K	Name	Name of the measure	String	255
K	Target	Name of the target	String	255
	Formula	Formula to assign to the selected measure and target combination	Formula	

Scorecard

Use this behaviour to import or export scorecards.

Table 20 Scorecard

Required Fields and Key	Name	Description	Type	Size
R	Action	C, V, U, D	Constant	
K	Name	Name of the object to which the scorecard is attached	String	255
K	Type	Type of object to which the scorecard is attached: <ul style="list-style-type: none"> ● Employee ● AME ● SME 	Constant	
	ID	Object ID	Integer	
K	Scorecard Name	Name of the scorecard	String	255
	Primary	If it is primary, the name of the scorecard	String	255
	Template	If the scorecard is derived from a template, the name of the template	String	255
	Template Weight	Amount of weight applied to items on the template (0.0 – 1.0)	Decimal	
	Context	Members that comprise the dimensional context of the scorecard. Missing dimensions default to “top”.	0–n (Dimension Numbers)	
	Categories	Categories to which the scorecard is assigned	0–n (Scorecard Categories)	
	Key	Primary Keys	String. The value must be delimited within \${}. For example, \${32f.-1.-f419e0}.	
	ID	Object ID	Integer	
K	Scorecard Name	Name of any secondary scorecards	String	255
	Date	Date the Named Range is in effect for the scorecard	Date	
	Named Range	Named range assigned to the scorecard for the date specified	0–1 (Named Range)	

ScorecardElements

Use this behaviour to import or export scorecard elements.

Table 21 ScorecardElements

Required Fields and Key	Name	Description	Type	Size
R	Action	C, V, U, D	Constant	
K	Name	Name of the object to which the scorecard is attached	String	255
K	Type	Type of object to which the scorecard is attached: <ul style="list-style-type: none"> ● Employee ● AME ● SME 	Constant	
	ID	Object ID	Integer	
K	Item Name	Name of item to attach to a scorecard	1-1 (Employee, AME, SME)	
K	Item Type	Type of object to add to a scorecard: Employee, AME, SME, or Measure	Constant	
K	Order	Sequential position of object as displayed on the scorecard	Integer	
K	Weight	Weighting of the object as displayed on the scorecard.	Decimal	
	Trend	Localized description of the trend	Constant	
	<Target> Score	Total scorecard score based on the target used to establish performance	Decimal	
	<Target>Weighted Score	Total scorecard score based on the weighting of the target	Decimal	
	<Target> Status	Name of the status symbol that indicates the level or scorecard performance. Performance evaluated using the target specified.	String	

StrategyMap

Use this behaviour to import or export Strategy maps.

Table 22 StrategyMap

Required Fields and Key	Name	Description	Type	Size
R	Action	C, V, U, D	Constant	
K	Name	Name of the Strategy Map	String	255
	Key	Primary Key	String The value must be delimited within \${}. For example \${32f.-1.-f419e0}.	
	ID	Object ID	Integer	
K	Description	Description of the purpose of the Strategy Map	String	2000
	Domains	Domains associated with the strategy map	0-n (Domain)	

StrategyMapElement

Use this behaviour to import and export strategy map elements.

Table 23 StrategyMapElement

Required Fields and Key	Name	Description	Type	Size
R	Action	C, V, U, D	Constant	
K	Name	Name of the Map element	String	255
	Key	Primary Key String The value must be delimited within \${}. For example \${32f.-1.- f419e0}.	String: The value must be delimited within \${}. For example \${32f.-1.- f419e0}.	
	ID	Object ID	Integer	
	Description	Purpose of the Strategy Map	String	2000
R	Map Name	Identifies the map on which the element is used	1-1 (Strategy Map)	
R	Parent	Parent map element. If blank, the element is at the root level of the map.	0-1 (SME)	
R	Order	Sequential order of the map element, relative to other map elements that	Integer	

Required Fields and Key	Name	Description	Type	Size
		are children of the same map element.		
	Lable		String	255
	Title	Title for the map element. Overrides the default title, which is based on level in the map hierarchy.	String	255
	Owners	Employee who owns the SME	0-n (Employee)	
	Annotation Creator	Employee who is an annotation creator for this SME	0-n (Employee)	
	Perspective Primary	Most logical perspective associated with the SME	0-1 (Perspective)	
	Secondary Perspectives	Another perspective with which the SME can be associated	0-n (Perspective)	
	Primary theme	Primary theme associated with the SME	0-1 (Strategic Theme)	
	Secondary Theme	Secondary theme associated with the SME	0-n (Strategic Theme)	

Target

Use this behaviour to import or export defined targets, but not target values.

Table 24 Target

Required Fields and Key	Name	Description	Type	Size
R	Action	C, V, U, D	Constant	
K	Name	Name of the Unit	String	255
	Key	Primary Key	String The value must be delimited within \${}. For example \${32f.-1.-f419e0}.	
	ID	Object ID	Integer	
	Description	Description of the purpose of the Strategy Map	String	2000

Required Fields and Key	Name	Description	Type	Size
R	Order	Unique integer to specify sort order. The lowest number displays first, and is considered the default target	Integer	
R	Parent	Parent map element. If blank, the element is at the root level of the map.	0-1 (SME)	
R	Order	Sequential order of the map element, relative to other map elements that are children of the same map element.	Integer	

Unit

Use this behaviour to import or export units.

Table 25 Unit Metadata

Required Fields and Key	Name	Description	Type	Size
R	Action	C, V, U, D	Constant	
K	Name	Name of the unit	String	255
	Key	Primary Key	String The value must be delimited within \${}. For example \${32f.-1.-f419e0}.	
	ID	Object ID	Integer	
	Description	Description of the purpose of the unit	String	2000
	Currency Code	Valid currency code.	String	16
	Scaling Factor	Scaling factors used to import or export to Hyperion® System™ 9 Shared Services™ only: <ul style="list-style-type: none"> ● 1 ● 10 ● 100 ● 1000 ● 10000 ● 100000 	Constant	

Required Fields and Key	Name	Description	Type	Size
		● 1000000		

UserAccount

Use this behaviour to import and export user accounts.

Table 26 User Accounts

Required Fields and Key	Name	Description	Type	Size
R	Action	C, V, U, D	Constant	
K	UserID	User ID for the account	String	255
	Key	Primary Key	String The value must be delimited within \${}. For example \${32f.-1.-f419e0}.	
	ID	Object ID	Integer	
	Password	Password attached to the user account	String	255
	Roles	Security roles attached to the user account	0-n (Role)	16
	Employee	Employee associated with the user account	0-n (Employee)	
	PrimaryDomain	Primary domain associated with the user account	0-1 (Domain)	
	Groups	Groups associated with the user account	0-n (UserGroup)	

Variables

Use this behaviour to import or export variables.

Table 27 Variable Metadata

Required Fields and Key	Name	Description	Type	Size
R	Action	C, V, U, D	Constant	
K	Name	Name of the variable	String	255
	Key	Primary Key	String	

Required Fields and Key	Name	Description	Type	Size
			The value must be delimited within \${}. For example \${32f.-1.-f419e0}.	
	ID	Object ID	Integer	
	Description	Description of the variable	String	2000
	Result Collectors	Employee who is a result collector	0-n (Employee)	

VariableResult

Use this behaviour to import or export results for variables.

Table 28 Variable Results

Required Fields and Key	Name	Description	Type	Size
R	Action	C, V, U, D	Constant	
K	Name	Name of the variable	String	255
	Key	Primary Key	String The value must be delimited within \${}. For example \${32f.-1.-f419e0}.	
	ID	Object ID	Integer	
	Date	Dates for which the results were collected	Date	
	Result	Result value	Decimal	

COPYRIGHT NOTICE

Copyright 2007 Hyperion Solutions Corporation. All rights reserved. Hyperion Solutions Corporation. All rights reserved.

Hyperion, the Hyperion logo, Hyperion Solutions, Essbase XTD, Hyperion Analyzer, and Hyperion Central are registered trademarks or trademarks of Hyperion Solutions Corporation.

All other brand and product names are trademarks or registered trademarks of their respective holders.

No portion of this manual may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or information storage and retrieval systems, for any purpose other than the purchaser's personal use, without the express written permission of Hyperion Solutions Corporation.

Notice: The information contained in this document is subject to change without notice. Hyperion Solutions Corporation shall not be liable for errors contained herein or consequential damages in connection with the furnishing, performance, or use of this material.

Hyperion Solutions Corporation

5450 Great America Parkway

Santa Clara, California 95054

Printed in the U.S.A.