



**ORACLE® HYPERION ENTERPRISE
PERFORMANCE MANAGEMENT SYSTEM
ORACLE® HYPERION STRATEGIC FINANCE, FUSION
EDITION**

RELEASE 11.1.1.1.00

**HSF WEB SERVICES API DEVELOPER'S
GUIDE**

ORACLE®
ENTERPRISE PERFORMANCE
MANAGEMENT SYSTEM

Strategic Finance HSF Web Services API Developer's Guide, 11.1.1.1.00

Copyright © 1996, 2009, Oracle and/or its affiliates. All rights reserved.

Authors: EPM Information Development Team

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this software or related documentation is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable: U.S. GOVERNMENT RIGHTS: Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

This software is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications which may create a risk of personal injury. If you use this software in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy and other measures to ensure the safe use of this software. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software in dangerous applications.

This software and documentation may provide access to or information on content, products and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third party content, products and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third party content, products or services.

Contents

Chapter 1. HSF Web Services API Overview	5
Overview	5
Disclaimer	5
Compatible Programming Frameworks	5
Architecture	6
Files and Directories	6
HYPERION_HOME\StrategicFinance\Server\hsfwebservices	6
..\hsfwebservices\bin	6
Logging	7
Registry Settings	7
Deploying HSF Web Services	7
Test Pages	8
Example—Using the Test Pages	8
Accessing the HSF Web Services Test Page	8
Returning a SessionID	8
Returning a List of HSF Servers	9
Accessing an HSF Server	9
Chapter 2. HSF Web Services API	11
Class Diagram	11
HSFWebService	11
Public Constructors	11
HSFWebService	11
General Functions	12
EnumServers	12
Server Functions	13
EnumDatabases	13
EnumOpenedEntities	14
OpenServer	15
CloseServer	16
Database Functions	16
CloseAllEntities	17

CloseDatabase	18
EnumCubes	19
EnumEntities	19
OpenDatabase	21
Session Functions	22
CreateSession	22
CreateSessionFromToken	22
CloseSession	23
GetSessionInfo	24
HSFEntityWebService	25
Public Constructors	25
HSFEntityWebService	25
Public Functions	25
CreateEntity	25
OpenEntity	27
CloseEntity	28
EnumAccounts	29
EnumCustomDimensions	30
EnumCustomMembers	31
EnumScenarios	32
EnumTimePeriods	33
EnumMeasures	34
GetEntityDataCells	35
SetEntityDataCells	36
GetEntityLockInfo	38
ReleaseEntityLock	38
Calculate	39
ExportExtendedAnalytics	40
Chapter 3. Sample Code	43
C# in Active Server Pages	43
IncomeStatement.aspx	43

1

HSF Web Services API Overview

In This Chapter

Overview	5
Architecture.....	6
Files and Directories	6
Registry Settings	7
Deploying HSF Web Services	7
Test Pages	8

Overview

Oracle Hyperion Strategic Finance, Fusion Edition Web Services is C# API (Application Programming Interface) for accessing and interacting with entities on Strategic Finance servers. This guide is to help developers understand how to use Strategic Finance Web Services to create web-enabled software that accesses Strategic Finance data. It enables integration with other Oracle Hyperion Enterprise Performance Management System products, such as:

- Oracle Hyperion Financial Data Quality Management, Fusion Edition

Disclaimer

The HSF Web Services API is subject to change.

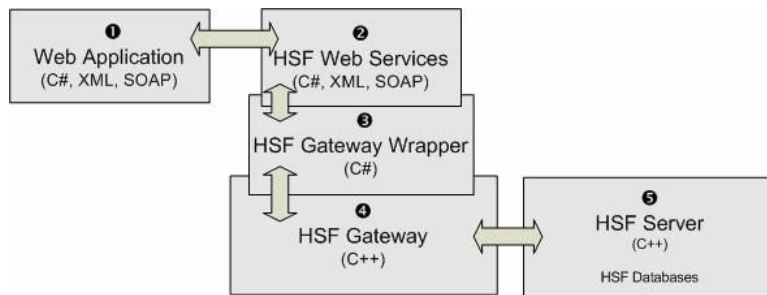
Compatible Programming Frameworks

The HSF Gateway is built with Microsoft .NET, so it is compatible with .NET frameworks, such as:

- Visual Basic
- C#
- C++
- JScript

Architecture

Figure 1 Architecture



Files and Directories

Strategic Finance Web Services is installed along with the Strategic Finance server. This section lists the directories and files they contain.

HYPERION_HOME\StrategicFinance\Server \hsfwebservices

Contains the following:

- HSFWebService.asmx
The web service
- HSFEntityWebServices.asmx
The entity web service
- web.config
Configuration file

..\hsfwebservices\bin

Contains the following:

- HSFGateway.dll
Contains the Gateway classes
- HSFGatewayCommunication.dll
Contains the gateway communication classes
- HSFGatewayWrapper.dll
Contains the gateway wrapper classes
- HSFWebServices.dll

Contains the HSF Web Services classes

- `Interop.HSFGatewayLib.dll`

Contains classes for use with Oracle's Hyperion® Shared Services and other Oracle Hyperion Enterprise Performance Management System products

Logging

For information on logging, search the web for using the `GetTempPath` function to locate the temporary logging directory.

Registry Settings

- file directory
Directory containing the gateway log file
- log
Determines whether the Gateway will log errors and actions
 - true
Log errors and actions
 - false
Do not log
- log file name
Filename of the Gateway log
- network address
HSF server IP address
- port
HSF server port
- protocol
Valid values:
 - `ncacn_ip_tcp`
- server name
Name of the HSFServer returned by `EnumServers`.
Maximum: 1

Deploying HSF Web Services

HSF Web Services are installed with the HSF server.

➤ To deploy HSF Web Services:

1 Install prerequisites:

- Microsoft Internet Information Services
- On Windows Server platforms, configure IIS to allow Active Server Pages.

2 Install HSF Server:

- Choose Custom instead of typical and make sure Web API Service is checked.
- If it reboots in the middle of the install, it may need to be re-installed.

Test Pages

HSF Web Services provides web pages you can use in testing and learning the API. These pages are accessible only on the local machine. You can access these pages using a browser on the local machine to access the following URLs:

HSF Web Services Test Page

`http://localhost/HSFWebServices/HSFWebService.asmx`

HSF Entity Web Services Test Page

`http://localhost/HSFWebServices/HSFEntityWebService.asmx`

These pages provides links showing most of the functions available, the parameters you can pass, and an example of the SOAP request and response. In addition, you can enter parameters and submit the request to HSF Web Services in order to obtain a response and see how the code behaves.

Example—Using the Test Pages

Accessing the HSF Web Services Test Page

➤ To access the test pages:

- 1 On the local machine running HSF Web Services, open a browser.**
- 2 Access the following URL:**

`http://localhost/HSFWebServices/HSFWebService.asmx`

Returning a SessionID

You will need a SessionID to perform all of the functions in the test pages.

- To return a SessionID:
 - 1 From the HSF Web Services test page, click **Create Session**.
 - 2 In **userName**, enter a user.
 - 3 In **password**, enter a password.
 - 4 Click **Invoke**.

When you click Invoke, you will receive a SOAP response containing a string, which represents a SessionID. You can copy and paste this SessionID for use with all other function test pages.

Returning a List of HSF Servers

- To return a list of HSF Servers:
 - 1 From the HSF Web Services test page, click **EnumServers**.
 - 2 In **sessionID**, enter a sessionID.
 - 3 Click **Invoke**.

Accessing an HSF Server

- To access an HSF server:
 - 1 From the HSF Web Services test page, click **OpenServer**.
 - 2 In **sessionID**, enter a sessionID.
 - 3 In **server**, enter an HSF server name.
 - 4 Click **Invoke**.

For many functions, success is indicated by a blank page, meaning that no error has been returned, but there is no other return value to view, as with a successful server logon.

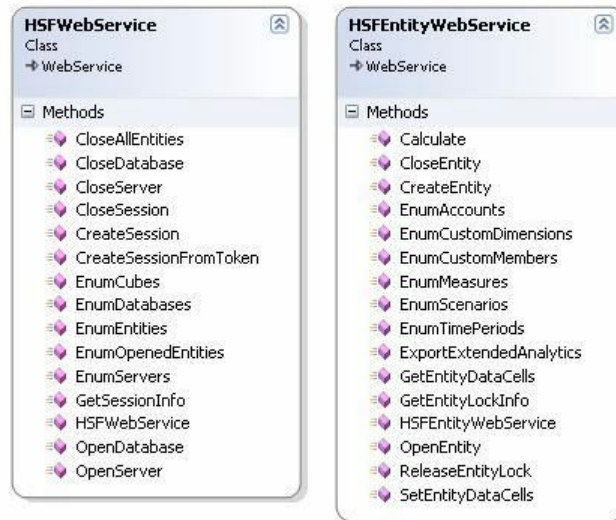
2

HSF Web Services API

In This Chapter

Class Diagram	11
HSFWebService	11
HSFEntityWebService	25

Class Diagram



HSFWebService

Contains functions for accessing Oracle Hyperion Strategic Finance, Fusion Edition servers and databases.

Public Constructors

HSFWebService

Initializes a new instance of the WebServices class.

Syntax

- **Visual Basic**

```
Public Sub New()
```

- **C#**

```
public HSFWebService()
```

- **C++**

```
public:  
    HSFWebService() sealed
```

- **JScript**

```
public function HSFWebService()
```

Parameters

None

Returns

HSFWebService object

Example

See [“IncomeStatement.aspx”](#) on page 43.

General Functions

EnumServers

Description: Returns a list of servers for the HSFWebService.

Pre-requisites: CreateSession()

Syntax

- **Visual Basic**

```
Public Function EnumServers( _  
    ByVal sessionID As String _  
) As String()
```

- **C#**

```
public string[] EnumServers(  
    string sessionID  
)
```

- **C++**

```
public:  
    array<>^ EnumServers(  
        String sessionID  
) sealed
```

- **JScript**

```
public function EnumServers(  
    sessionID : String  
    ) : String[]
```

Parameters

Table 1 EnumServers Parameters

Parameter	Data Type	Description
sessionID	string	A session identifier

Returns

A list of servers

Example

See [“IncomeStatement.aspx”](#) on page 43.

Server Functions

These functions access HSF servers.

EnumDatabases

Description: Returns a list of databases for the server currently open in the HSFWebService.

Pre-requisites: CreateSession() and OpenServer().

Syntax

- **Visual Basic**

```
Public Function EnumDatabases( _  
    ByVal sessionID As String _  
    ) As String()
```

- **C#**

```
public string[] EnumDatabases(  
    string sessionID  
    )
```

- **C++**

```
array<String>^ EnumDatabases(  
    String sessionID  
    ) sealed
```

- **JScript**

```
public function EnumDatabases(  
    sessionID : String  
    ) : String[]
```

Parameters

Table 2 EnumDatabases Parameters

Parameter	Data Type	Description
sessionID	string	A session identifier

Returns

HSFWebService object

Example

See [“IncomeStatement.aspx”](#) on page 43.

EnumOpenedEntities

Description: Enumerates entities opened in the current HSF server.

Pre-requisites: CreateSession()

```
public DataSet EnumOpenedEntities(string sessionID)
```

Syntax

- **Visual Basic**

```
Public Function EnumOpenedEntities( _  
    ByVal sessionID As String _  
) As DataSet
```

- **C#**

```
public DataSet EnumOpenedEntities(  
    string sessionID  
)
```

- **C++**

```
public:  
    DataSet EnumOpenedEntities(  
        String sessionID  
    ) sealed
```

- **JScript**

```
public function EnumOpenedEntities(  
    sessionID : String  
) : DataSet
```

Parameters

Table 3 EnumOpenedEntities Parameters

Parameter	Data Type	Description
sessionID	string	A session identifier

Returns

HSFWebService object

OpenServer

Description: Opens a connection to a specified HSF Server.

Pre-requisites: CreateSession(), OpenServer(), OpenDatabase() and OpenCube().

Syntax

- **Visual Basic**

```
Public Sub OpenServer( _  
    ByVal sessionID As String, _  
    ByVal server As String _  
)
```

- **C#**

```
public void OpenServer(  
    string sessionID,  
    string server  
)
```

- **C++**

```
public:  
    void OpenServer(  
        String sessionID,  
        String server  
    ) sealed
```

- **JScript**

```
public function OpenServer(  
    sessionID : String,  
    server : String  
)
```

Parameters

Table 4 OpenServer Parameters

Parameter	Data Type	Description
sessionID	string	A session identifier
server	string	An HSF server name

Returns

- void
 - Success
- Error message
 - Failure

Example

See [“IncomeStatement.aspx”](#) on page 43.

CloseServer

Description: Closes the server currently open in the HSFWebService.

Pre-requisites: CreateSession() and OpenServer().

Syntax

- **Visual Basic**

```
Public Sub CloseServer( _  
    ByVal sessionID As String _  
)
```

- **C#**

```
public void CloseServer(  
    string sessionID  
)
```

- **C++**

```
public:  
    void CloseServer(  
        String sessionID  
) sealed
```

- **JScript**

```
public function CloseServer(  
    sessionID : String  
)
```

Parameters

Table 5 CloseServer Parameters

Parameter	Data Type	Description
sessionID	string	A session identifier

Returns

- void
 - Success
- Error message
 - Failure

Database Functions

These functions access HSF databases.

CloseAllEntities

Description: Closes all entities in the database currently open in the HSFWebService.

Prerequisites: CreateSession(), OpenServer() and OpenDatabase().

Syntax

- **Visual Basic**

```
Public Sub CloseAllEntities( _  
    ByVal sessionID As String, _  
    ByVal checkIn As Boolean, _  
    ByVal releaseLock As Boolean _  
)
```

- **C#**

```
public void CloseAllEntities(  
    string sessionID,  
    bool checkIn,  
    bool releaseLock  
)
```

- **C++**

```
public:  
    void CloseAllEntities(  
        String sessionID,  
        bool checkIn,  
        bool releaseLock  
    ) sealed
```

- **JScript**

```
public function CloseAllEntities(  
    sessionID : String,  
    checkIn : bool,  
    releaseLock : bool  
)
```

Parameters

Table 6 CloseAllEntities Parameters

Parameter	Data Type	Description
sessionID	string	A session identifier
checkIn	bool	<ul style="list-style-type: none">● True—check in entity● False—do not check in
releaseLock	bool	<ul style="list-style-type: none">● True—release lock● False—retain lock

Returns

- void

- Success
- Error message
- Failure

CloseDatabase

Description: Closes the database currently open in the HSFWebService.

Prerequisites: CreateSession(), OpenServer() and OpenDatabase().

Syntax

- **Visual Basic**

```
Public Sub CloseDatabase( _
    ByVal sessionID As String _
)
```

- **C#**

```
public void CloseDatabase(
    string sessionID
)
```

- **C++**

```
public:
    void CloseDatabase(
        String sessionID
    ) sealed
```

- **JScript**

```
public function CloseDatabase(
    sessionID : String
)
```

Parameters

Table 7 CloseDatabase Parameters

Parameter	Data Type	Description
sessionID	string	A session identifier

Returns

- void
- Success
- Error message
- Failure

EnumCubes

Description: Returns a list of cubes for the database currently open in the HSFWebService.

Prerequisites: CreateSession(), OpenServer() and OpenDatabase().

Syntax

- **Visual Basic**

```
Public Function EnumCubes( _  
    ByVal sessionID As String _  
) As String()
```

- **C#**

```
public string[] EnumCubes(  
    string sessionID  
)
```

- **C++**

```
public:  
    array<String>^ EnumCubes(  
        String sessionID  
) sealed
```

- **JScript**

```
public function EnumCubes(  
    sessionID : String  
) : String[]
```

Parameters

Table 8 EnumCubes Parameters

Parameter	Data Type	Description
sessionID	string	A session identifier

Returns

HSFWebService object

EnumEntities

Description: Returns a list of entities for the database currently open in the HSF Gateway. If an entity is specified, the list will contain only ancestors of that entity.

Prerequisites: CreateSession(), OpenServer() and OpenDatabase().

Syntax

- **Visual Basic**

```
Public Function EnumEntities( _  
    ByVal sessionID As String, _  
    ByVal entity As String, _
```

```

        ByVal action As String _
    ) As String()

```

- **C#**

```

public string[] EnumEntities(
    string sessionID,
    string entity,
    string action
)

```

- **C++**

```

public:
    array<String>^ EnumEntities(
        String sessionID,
        String entity,
        String action
    ) sealed

```

- **JScript**

```

public function EnumEntities(
    sessionID : String,
    entity : String,
    action : String
) : String[]

```

Parameters

Table 9 EnumEntities Parameters

Parameter	Data Type	Description
sessionID	string	A session identifier
entity	string	An entity nameAn empty string (“”) indicates starting at the root.
action	string	Use one of the following: <ul style="list-style-type: none"> ● <blank> (default) or all Enumerates all entities ● children Enumerates down to the children ● descendants Enumerates down to descendants

Returns

A list of entities in the open database

Example

See [“IncomeStatement.aspx”](#) on page 43.

OpenDatabase

Description: Opens a database in the server currently open in the HSFWebService.

Prerequisites: CreateSession() and OpenServer().

Syntax

- **Visual Basic**

```
Public Sub OpenDatabase( _  
    ByVal sessionID As String, _  
    ByVal database As String _  
)
```

- **C#**

```
public void OpenDatabase(  
    string sessionID,  
    string database  
)
```

- **C++**

```
public:  
    void OpenDatabase(  
        String sessionID,  
        String database  
    ) sealed
```

- **JScript**

```
public function OpenDatabase(  
    sessionID : String,  
    database : String  
)
```

Parameters

Table 10 OpenDatabase Parameters

Parameter	Data Type	Description
sessionID	string	A session identifier
database	string	A database name

Returns

- void
 - Success
- Error message
 - Failure

Example

See [“IncomeStatement.aspx”](#) on page 43.

Session Functions

Currently unavailable.

CreateSession

Description: Creates a session for the given user/password on the HSFWebServices.

Prerequisites: None.

Syntax

- Visual Basic
- C#

```
public string CreateSession(string userName, string password)
```

- C++
- JScript

Parameters

Table 11 CreateSession Parameters

Parameter	Data Type	Description
userName	string	A user name
password	string	The user's password

Returns

A session string

Example

See [“IncomeStatement.aspx”](#) on page 43.

CreateSessionFromToken

Description: Creates a session for the given Oracle's Hyperion® Shared Services Single Sign-on token on the HSFWebServices.

Prerequisites: None.

Syntax

- Visual Basic
- C#

```
public string CreateSessionFromToken(string token)
```

- C++

- JScript

Parameters

Table 12 Parameters

Parameter	Data Type	Description
token	string	A CSS token

Returns

A session string

Example

See .

CloseSession

Description: Closes the given session currently open in the HSFWebService.

Prerequisites: CreateSession().

Syntax

- Visual Basic

```
Public Sub CloseSession( _  
    ByVal sessionID As String _  
)
```

- C#

```
public void CloseSession(  
    string sessionID  
)
```

- C++

```
public:  
    void CloseSession(  
        String sessionID  
) sealed
```

- JScript

```
public function CloseSession(  
    sessionID : String  
)
```

Parameters

Table 13 CloseSession Parameters

Parameter	Data Type	Description
sessionID	string	A session identifier

Returns

- void
Success
- Error message
Failure

Example

See .

GetSessionInfo

Description: Retrieves information about the user session including token.

Prerequisites: CreateSession().

Syntax

- **Visual Basic**

```
Public Function GetSessionInfo( _  
    ByVal sessionID As String _  
) As DataSet
```
- **C#**

```
public DataSet GetSessionInfo(  
    string sessionID  
)
```
- **C++**

```
public:  
    DataSet GetSessionInfo(  
        String sessionID  
) sealed
```
- **JScript**

```
public function GetSessionInfo(  
    sessionID : String  
) : DataSet
```


Parameters

Table 14 GetSessionInfo Parameters

Parameter	Data Type	Description
sessionID	string	A session identifier

Returns

void

HSFEntityWebService

Public Constructors

HSFEntityWebService

Creates an instance of an entity web services, for accessing HSF entities.

Syntax

- Visual Basic
- C#

```
public HSFEntityWebService()
```
- C++
- JScript

Parameters

None

Returns

An HSFEntityWebService object

Public Functions

CreateEntity

Description: Creates a new entity. It can be placed in a structure by optionally including root and parent entity names. There is a flag to retain data from within the source entity in the new entity.

Prerequisites: CreateSession(), OpenServer() and OpenDatabase().

Syntax

- **Visual Basic**

```
Public Function CreateSession( _  
    ByVal sessionID As String, _  
    ByVal userName As String, _  
    ByVal password As String, _  
    ByVal token As String _  
    ) As String
```

- **C#**

```
public string CreateSession(  
    string sessionID,  
    string userName,  
    string password,  
    string token  
    )
```

- **C++**

```
public:  
    String CreateSession(  
        String sessionID,  
        String userName,  
        String password,  
        String token  
    ) sealed
```

- **JScript**

```
public function CreateSession(  
    sessionID : String,  
    userName : String,  
    password : String,  
    token : String  
    ) : String
```

Parameters

Table 15 CreateSession Parameters

Parameter	Data Type	Description
sessionID	string	A session identifier
newEntity	string	A name for the new entity
baseEntity	string	A source entity from which to copy the new entity
rootEntity	string	The root entity of the tree
parentEntity	string	The entity under which to create the new entity
keepData	bool	Indicates whether or not to copy the data from the baseEntity

Returns

void

OpenEntity

Description: Opens an entity in the database currently open in the HSFWebService.

Prerequisites: CreateSession(), OpenServer() and OpenDatabase().

Syntax

- **Visual Basic**

```
Public Function OpenEntity( _  
    ByVal sessionID As String, _  
    ByVal entity As String, _  
    ByVal checkOut As Boolean _  
    ) As DataSet
```

- **C#**

```
public DataSet OpenEntity(  
    string sessionID,  
    string entity,  
    bool checkOut  
    )
```

- **C++**

```
public:  
    DataSet OpenEntity(  
        String sessionID,  
        String entity,  
        bool checkOut  
    ) sealed
```

- **JScript**

```
public function OpenEntity(  
    sessionID : String,  
    entity : String,  
    checkOut : bool  
    ) : DataSet
```

Parameters

Table 16 Parameters

Parameter	Data Type	Description
sessionID	string	A session identifier
entity	string	An entity name
checkOut	bool	<ul style="list-style-type: none">● True—check out entity● False—open as copy

Returns

DataSet

Table 17 OpenEntity DataSet

Parameter	Data Type	Description
Name	String	Name of entity
IsCheckedOut	boolean	Checked out status: <ul style="list-style-type: none">● true Is checked out● false Is not checked out

Example

See [“IncomeStatement.aspx”](#) on page 43.

CloseEntity

Description: Closes the entity currently open in the HSFWebService.

Prerequisites: CreateSession(), OpenServer(), OpenDatabase() and OpenEntity().

Syntax

- **Visual Basic**

```
Public Sub CloseEntity( _  
    ByVal sessionID As String, _  
    ByVal checkIn As Boolean, _  
    ByVal releaseLock As Boolean _  
)
```

- **C#**

```
public void CloseEntity(  
    string sessionID,  
    bool checkIn,  
    bool releaseLock  
)
```

- **C++**

```
public:  
    void CloseEntity(  
        String sessionID,  
        bool checkIn,  
        bool releaseLock  
    ) sealed
```

- **JScript**

```
public function CloseEntity(  
    sessionID : String,  
    checkIn : bool,  
    releaseLock : bool  
)
```

Parameters

Table 18 CloseEntity Parameters

Parameter	Data Type	Description
sessionID	string	A session identifier
checkIn	bool	<ul style="list-style-type: none">● True—check in entity● False—do not check in
releaseLock	bool	<ul style="list-style-type: none">● True—release lock● False—retain lock

Returns

- void
Success
- Error message
Failure

EnumAccounts

Description: Returns a list of accounts for the entity currently open in the HSFWebService.

Prerequisites: CreateSession(), OpenServer(), OpenDatabase() and OpenEntity().

Syntax

- **Visual Basic**

```
Public Function EnumAccounts( _  
    ByVal sessionID As String _  
) As DataSet
```

- **C#**

```
public DataSet EnumAccounts(  
    string sessionID  
)
```

- **C++**

```
public:  
    DataSet EnumAccounts(  
        String sessionID  
) sealed
```

- **JScript**

```
public function EnumAccounts(  
    sessionID : String  
) : DataSet
```

Parameters

Table 19 EnumAccounts Parameters

Parameter	Data Type	Description
sessionID	string	A session identifier

Returns

DataSet

Table 20 EnumAccounts DataSet

Parameter	Data Type	Description
ID	string	Account ID
Name	string	Account name
Number	string	Account number
IsInput	boolean	Indicates if the account is an input or an output: <ul style="list-style-type: none">● true Input account● false Output account

Example

See [“IncomeStatement.aspx”](#) on page 43.

EnumCustomDimensions

Syntax

- Visual Basic
- C#

```
public DataSet EnumCustomDimensions(  
    string sessionID  
)
```

- C++
- JScript

Parameters

None

Returns

DataSet

Table 21 EnumCustomDimensions DataSet

Parameter	Data Type	Description
Name	string	Custom dimension name

Example

See .

EnumCustomMembers

Syntax

- Visual Basic
- C#

```
public DataSet EnumCustomMembers(
    string sessionID,
    string dimension,
    string parent,
    string action
)
```

- C++
- JScript

Parameters

None

Returns

DataSet

Table 22 EnumCustomMembers DataSet

Parameter	Data Type	Description
ID	string	Custom member ID
ParentID	string	Pointer to parent custom member, if it exists
Name	string	Custom member name
IsParent	bool	Indicates if the custom member is a parent: <ul style="list-style-type: none"> • true Is parent with children • false Has no children
IsInput	bool	Indicates if ?:

Parameter	Data Type	Description
		<ul style="list-style-type: none"> ● true ? ● false ?

EnumScenarios

Description: Returns a list of scenarios for the entity currently open in the HSF Gateway.

Prerequisites: CreateSession(), OpenServer(), OpenDatabase() and OpenEntity().

Syntax

- **Visual Basic**

```
Public Function EnumScenarios( _
    ByVal sessionID As String _
) As DataSet
```

- **C#**

```
public DataSet EnumScenarios(
    string sessionID
)
```

- **C++**

```
public:
    DataSet EnumScenarios(
        String sessionID
    ) sealed
```

- **JScript**

```
public function EnumScenarios(
    sessionID : String
) : DataSet
```

Parameters

Table 23 Parameters

Parameter	Data Type	Description
sessionID	string	A session identifier

Returns

DataSet

Table 24 EnumScenarios DataSet

Parameter	Data Type	Description
ID	string	Scenario ID
Name	string	Scenario name

Example

See [“IncomeStatement.aspx”](#) on page 43.

EnumTimePeriods

Description: Returns a list of time periods for the entity currently open in the HSFWebService.

Prerequisites: CreateSession(), OpenServer(), OpenDatabase() and OpenEntity().

Syntax

- **Visual Basic**

```
Public Function EnumTimePeriods( _
    ByVal sessionID As String _
) As DataSet
```

- **C#**

```
public DataSet EnumTimePeriods(
    string sessionID
)
```

- **C++**

```
public:
    DataSet EnumTimePeriods(
        String sessionID
    ) sealed
```

- **JScript**

```
public function EnumTimePeriods(
    sessionID : String
) : DataSet
```

Parameters

Table 25 EnumTimePeriods Parameters

Parameter	Data Type	Description
sessionID	string	A session identifier

Returns

DataSet

Table 26 EnumTimePeriods DataSet

Parameter	Data Type	Description
ID	string	Time Period ID
ParentID	string	Pointer to parent Time Period, if it exists
Name	string	Time Period name
IsParent	bool	Indicates if the time period is a parent: <ul style="list-style-type: none"> ● true Is parent with children ● false Has no children
IsInput	bool	Indicates if ? : <ul style="list-style-type: none"> ● true ? ● false ?

Example

See [“IncomeStatement.aspx”](#) on page 43.

EnumMeasures

Description: Returns a list of measures, whether the value is an input or an output in the entity, for the entity currently open in the HSFWebService.

Prerequisites: CreateSession(), OpenServer(), OpenDatabase() and OpenEntity().

Syntax

- **Visual Basic**

```
Public Function EnumMeasures( _
    ByVal sessionID As String _
) As DataSet
```

- **C#**

```
public DataSet EnumMeasures(
    string sessionID
)
```

- **C++**

```
DataSet EnumMeasures(
    String sessionID
) sealed
```

- **JScript**

```

public function EnumMeasures(
    sessionID : String
) : DataSet

```

Parameters

Table 27 Parameters

Parameter	Data Type	Description
sessionID	string	A session identifier

Returns

DataSet

Table 28 EnumMeasures DataSet

Parameter	Data Type	Description
ID	string	Measure ID
Name	string	Measure name

Example

See .

GetEntityDataCells

Description: Returns a DataSet containing information from cells in the entity currently open in the HSFWebService.

Prerequisites: CreateSession(), OpenServer(), OpenDatabase() and OpenEntity().

Syntax

- **Visual Basic**

```

Public Function GetEntityDataCells( _
    ByVal sessionID As String, _
    ByVal dataCellLocators As DataSet _
) As DataSet

```

- **C#**

```

public DataSet GetEntityDataCells(
    string sessionID,
    DataSet dataCellLocators
)

```

- **C++**

```

public:
    DataSet GetEntityDataCells(
        String sessionID,

```

```

        DataSet dataCellLocators
    ) sealed

```

- **JScript**

```

public function GetEntityDataCells(
    sessionID : String,
    dataCellLocators : DataSet
) : DataSet

```

Parameters

Table 29 GetEntityDataCells Parameters

Parameter	Data Type	Description
sessionID	string	A session identifier
dataCellLocators	DataSet	A list of cell locations mapping to a DataTable

Returns

DataSet

Table 30 GetEntityDataCells DataSet

Parameter	Data Type	Description
Value	string	Value in cell
Account	string	Account row for cell
Time	string	Time period of cell
Scenario	string	Scenario of cell
Measure	string	Measure of cell
IsValid	bool	??
CanRead	bool	??
CanWrite	bool	??

Example

See [“IncomeStatement.aspx”](#) on page 43.

SetEntityDataCells

Description: Sends data to the entity currently open in the HSFWebService.

Prerequisites: CreateSession(), OpenServer(), OpenDatabase() and OpenCube().

Syntax

- Visual Basic

```
Public Sub SetEntityDataCells( _  
    ByVal sessionID As String, _  
    ByVal dataCellInfos As DataSet _  
)
```

- C#

```
public void SetEntityDataCells(  
    string sessionID,  
    DataSet dataCellInfos  
)
```

- C++

```
public:  
    void SetEntityDataCells(  
        String sessionID,  
        DataSet dataCellInfos  
    ) sealed
```

- JScript

```
public function SetEntityDataCells(  
    sessionID : String,  
    dataCellInfos : DataSet  
)
```

Parameters

Table 31 GetEntityDataCells Parameters

Parameter	Data Type	Description
sessionID	string	A session identifier
dataCellInfos	DataSet	A list of cell locations mapping to a DataTable

Returns

DataSet

Table 32 GetEntityDataCells DataSet

Parameter	Data Type	Description
Value	string	Value in cell
Account	string	Account row for cell
Time	string	Time period of cell
Scenario	string	Scenario of cell
Measure	string	Measure of cell
CustomMembers	string	??

Parameter	Data Type	Description
IsValid	bool	??
Cause	string	??

Example

See .

GetEntityLockInfo

Returns the name of the user holding the lock on the entity.

Syntax

- Visual Basic

- C#

```
public string GetEntityLockInfo(
    string sessionID,
    string Entity
)
```

- C++

- JScript

Parameters

Table 33 GetEntityLockInfo Parameters

Parameter	Data Type	Description
sessionID	string	A session identifier
Entity	string	Name of entity

Returns

string

ReleaseEntityLock

Syntax

- Visual Basic

- C#

```
public bool ReleaseEntityLock(
    string sessionID,
    string entity
)
```

- C++
- JScript

Parameters

Table 34 ReleaseEntityLock Parameters

Parameter	Data Type	Description
sessionID	string	A session identifier
Entity	string	Name of entity

Returns

void

Example

See [“IncomeStatement.aspx”](#) on page 43.

Calculate

Description: Calculates the entity currently open in the HSFWebService.

Prerequisites: CreateSession(), OpenServer(), OpenDatabase() and OpenEntity().

Syntax

- Visual Basic

```
Public Sub Calculate( _
    ByVal sessionID As String, _
    ByRef normalCalc As Boolean, _
    ByRef reverseCalc As Boolean _
)
```

- C#

```
public void Calculate(
    string sessionID,
    bool normalCalc,
    bool reverseCalc
)
```

- C++

```
public:
    void Calculate(
        String sessionID,
        bool% normalCalc,
        bool% reverseCalc
    ) sealed
```

- JScript

```
public function Calculate(
    sessionID : String,
```

```

        normalCalc : bool,
        reverCalc : bool
    )

```

Calculate Parameters

Table 35 Parameters

Parameter	Data Type	Description
sessionID	string	A session identifier
normalCalc	boolean	If true, performs a normal calculation.
reverseCalc	boolean	If true, performs a reverse calculation. Reverse calculations are performed first.

Returns

DataSet

Table 36 EnumMeasures DataSet

Parameter	Data Type	Description
Message	string	The message
IsError	bool	??? <ul style="list-style-type: none"> ● true Message is an error ● false Message is not an error

ExportExtendedAnalytics

Description: Does an extended analytics export for the entity currently open in the HSFWebService.

Prerequisites: CreateSession(), OpenServer(), OpenDatabase() and OpenEntity().

Syntax

- **Visual Basic**

```

Public Sub ExportExtendedAnalytics( _
    ByVal sessionID As String, _
    ByVal analysisVersion As String _
)

```

- **C#**

```

public void ExportExtendedAnalytics(
    string sessionID,
    string analysisVersion
)

```


- C++

```
public:  
    void ExportExtendedAnalytics(  
        String sessionID,  
        String analysisVersion  
    ) sealed
```

- JScript

```
public function ExportExtendedAnalytics(  
    sessionID : String,  
    analysisVersion : String  
)
```

Parameters

Table 37 ExportExtendedAnalytics Parameters

Parameter	Data Type	Description
sessionID	string	A session identifier
analysisVersion	string	An Extended Analytics name

Returns

- void
 - Success
- Error message
 - Failure

In This Chapter

C# in Active Server Pages43

C# in Active Server Pages

IncomeStatement.aspx

```
<%@ Page Language="C#" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://
www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<xml runat="server" id="dimensions">
  <dimensions>
    <entities>
      <entity>1</entity>
    </entities>
    <scenarios>All</scenarios>
    <measures>
      <measure>Output</measure>
    </measures>
    <accounts>
      <account>1030.00.000</account>
      <account>1040.00.000</account>
      <account>-</account>
      <account>1070.00.000</account>
      <account></account>
      <account>1080.00.010</account>
      <account>1080.00.020</account>
      <account>1080.00.030</account>
      <account>-</account>
      <account>1080.00.000</account>
      <account>1115.00.000</account>
      <account>1110.00.000</account>
      <account>-</account>
      <account>1150.00.000</account>
      <account></account>
      <account>1160.00.000</account>
      <account>1170.00.000</account>
      <account>-</account>
      <account>1200.00.000</account>
    </accounts>
  </dimensions>
</xml>
```

```

        <account></account>
        <account>1240.00.000</account>
        <account>1420.00.000</account>
        <account>-</account>
        <account>1600.00.000</account>
        <account>1690.00.000</account>
        <account>-</account>
        <account>1700.00.000</account>
        <account></account>
        <account>1730.00.000</account>
        <account></account>
        <account>1750.00.000</account>
        <account>=</account>
        <account></account>
        <account>7027.00.000</account>
        <account></account>
        <account>1800.00.000</account>
        <account>-</account>
        <account>1850.00.000</account>
        <account></account>
        <account>1900.00.000</account>
        <account></account>
        <account></account>
    </accounts>
    <time>All</time>
</dimensions>
</xml>
<script runat="server">
    protected void Page_Load(object sender, System.EventArgs eventArgs)
    {
        string user = Request.Params["user"];
        if (user == null || user == "") user = "username";
        string password = Request.Params["password"];
        if (password == null || password == "") password = "password";
        string hsfServer = Request.Params["hsfServer"];
        if (hsfServer == null || hsfServer == "") hsfServer = "HSFServer";
        string database = Request.Params["database"];
        string entity = Request.Params["entity"];
        if (entity == null || entity == "") entity = "1";
        string scenario = Request.Params["scenario"];
        if (scenario == null || scenario == "") scenario = "Base";
        string measure = Request.Params["measure"];
        if (measure == null || measure == "") measure = "Output";

        bool entityOpened = false;
        string sessionID = "";
        HSFEntityWebService.HSFEntityWebService hsfEntityWebService = null;

        try
        {
            // Create general web service link.
            HSFWebService.HSFWebService hsfWebService = new
HSFWebService.HSFWebService();

            // Create session with user/password.
            sessionID = hsfWebService.CreateSession(user, password);
            if (sessionID == "") throw new Exception("Can not create a

```

```

valid session.");

        // Open server.
        hsfWebService.OpenServer(sessionID, hsfServer);

        // Enum databases and make sure there is at least one.
        string[] databases = hsfWebService.EnumDatabases(sessionID);
        if (databases.Length < 1) throw new Exception("There are no
databases in the current server.");

        // Open database.
        if (database == null || database == "") database = databases
[0];

        hsfWebService.OpenDatabase(sessionID, database);

        // Get a list of entities.
        System.Data.DataSet entities = hsfWebService.EnumEntities
(sessionID, "", "All");

        if(entities == null || entities.Tables["Entities"] == null)
throw new Exception("There are no entities in this database");
        System.Data.DataRow[] entityRows = entities.Tables
["Entities"].Select();
        if (entityRows.Length < 1) throw new Exception("There are no
entities in this database");
        entity = (string)entityRows[0]["ID"];

        // Create entity web service link.
        hsfEntityWebService = new
HSFEntityWebService.HSFEntityWebService();

        // Open entity.
        hsfEntityWebService.OpenEntity(sessionID, entity, false);

        // Mark entity as opened.
        entityOpened = true;

        // Get list of time periods.
        System.Data.DataSet timePeriods =
hsfEntityWebService.EnumTimePeriods(sessionID);
        if(timePeriods == null || timePeriods.Tables["TimePeriods"] ==
null) throw new Exception("There are no time periods in this entity");
        System.Data.DataRow[] timePeriodRows = timePeriods.Tables
["TimePeriods"].Select();
        if (timePeriodRows.Length < 1) throw new Exception("here are no
time periods in this entity");
        int timeCols = timePeriodRows.Length;

        System.Data.DataSet scenarios =
hsfEntityWebService.EnumScenarios(sessionID);
        if (scenarios == null || scenarios.Tables["Scenarios"] == null)
throw new Exception("There are no scenarios in this entity");
        System.Data.DataRow[] scenarioRows = scenarios.Tables
["Scenarios"].Select();
        if (scenarioRows.Length < 1) throw new Exception("here are no
scenarios in this entity");

```

```

        System.Data.DataSet accounts = hsfEntityWebService.EnumAccounts
(sessionID);
        if (accounts == null || accounts.Tables["Accounts"] == null)
throw new Exception("There are no accounts in this entity");
        System.Data.DataTable accountTable = accounts.Tables
["Accounts"];
        System.Data.DataRow[] accountRows = accountTable.Select();
        if (accountRows.Length < 1) throw new Exception("here are no
accounts in this entity");

        string[] accountList = null;
        string xml = this.dimensions.InnerText;
        System.Xml.XmlDocument xmlDoc = new System.Xml.XmlDocument();
        xmlDoc.LoadXml(xml);
        System.Xml.XmlNodeList accountNodes = xmlDoc.SelectNodes
("descendant::dimensions/accounts/account");
        if (accountNodes != null && accountNodes.Count > 0)
        {
            accountList = new string[accountNodes.Count];
            for (int j = 0; j < accountNodes.Count; j++)
            {
                // Get node and create a new data cell info object.
                System.Xml.XmlNode accountNode = accountNodes.Item(j);
                if (accountNode != null)
                {
                    accountList[j] = accountNode.InnerText;
                    if (accountList[j] == "") accountList[j] = " ";
                }
            }
        }

        System.Data.DataSet dataCellsOut = null;
        // Create data set for GetCellData call.
        System.Data.DataSet dataCellsIn = new System.Data.DataSet();
        System.Data.DataTable dataCellsInTable = dataCellsIn.Tables.Add
("DataCells");

        // Add columns to the table.
        dataCellsInTable.Columns.Add("Entity", typeof(string));
        dataCellsInTable.Columns.Add("Account", typeof(string));
        dataCellsInTable.Columns.Add("Time", typeof(string));
        dataCellsInTable.Columns.Add("Scenario", typeof(string));
        dataCellsInTable.Columns.Add("Measure", typeof(string));
        dataCellsInTable.Columns.Add("CustomMembers", typeof(string));

        foreach (string account in accountList)
        {
            for (int j = 0; j < timeCols; j++)
            {
                System.Data.DataRow dataRow = dataCellsInTable.NewRow

                dataRow["Entity"] = entity;
                dataRow["Account"] = account;
                dataRow["Measure"] = measure;
                dataRow["Scenario"] = scenario;
                dataRow["Time"] = timePeriodRows[j]["ID"];
                dataRow["CustomMembers"] = "";
            }
        }
    }
}

```

```

        dataCellsInTable.Rows.Add(dataRow);
    }
}

// Call GetDataCells.
dataCellsOut = hsfEntityWebService.GetEntityDataCells
(sessionID, dataCellsIn);
if (dataCellsOut == null) throw new Exception("DataCells data
set not found.");

System.Data.DataTable dataCellsOutTable = dataCellsOut.Tables
["DataCells"];
if (dataCellsOutTable == null) throw new Exception("DataCells
table not found.");

// Create table.
HtmlTable table = new HtmlTable();
HtmlTableRow row;
HtmlTableCell cell;

// Create the top header row.
row = new HtmlTableRow();

// Create the first two header cells and add them to the
current row.
cell = new HtmlTableCell("th");
row.Cells.Add(cell);
cell = new HtmlTableCell("th");
cell.Align = "center";
cell.InnerHtml = "Account Names";
row.Cells.Add(cell);

// Create the time header.
int i;
for (i = 0; i < timeCols; i++)
{
    System.Data.DataRow timePeriodRow = timePeriodRows[i];

    cell = new HtmlTableCell("th");
    cell.InnerHtml = (string)timePeriodRow["Name"];
    row.Cells.Add(cell);
}

// Add the row to the table.
table.Rows.Add(row);

// Create the next header row and add it to the table;
row = new HtmlTableRow();
cell = new HtmlTableCell("th");
row.Cells.Add(cell);
cell = new HtmlTableCell();
cell.Align = "left";
cell.ColSpan = 4;
cell.InnerHtml = "File: 1.alc Last Calculated 9:38:05 PM 6/10/
2008";
row.Cells.Add(cell);
cell = new HtmlTableCell();

```

```

        cell.Align = "right";
        cell.ColSpan = timeCols - 3;
        string scenarioList = "Scenario:<select name=\"scenarioList\"
onChange=\"gotosite(this.options[this.selectedIndex].value)\">>";
        for (i = 0; i < scenarioRows.Length; i++)
        {
            scenarioList += "<option value=\"IncomeStatement.aspx?
scenario=" + (string)scenarioRows[i]["ID"] + "\"";
                if (scenario == (string)scenarioRows[i]["ID"]) scenarioList
+= " selected";
            scenarioList += ">" + scenarioRows[i]["ID"] + "</option>";
        }
        scenarioList += "</select>";
        cell.InnerHtml = scenarioList;
        row.Cells.Add(cell);
        table.Rows.Add(row);

// Create the next header row and add it to the table;
row = new HtmlTableRow();
cell = new HtmlTableCell("th");
row.Cells.Add(cell);
cell = new HtmlTableCell();
cell.Align = "center";
cell.ColSpan = timeCols + 1;
cell.InnerHtml = "Income Statement for Sample Incorporated";
row.Cells.Add(cell);
table.Rows.Add(row);

// Create the next header row and add it to the table;
row = new HtmlTableRow();
cell = new HtmlTableCell("th");
row.Cells.Add(cell);
cell = new HtmlTableCell();
cell.Align = "center";
cell.ColSpan = timeCols + 1;
cell.InnerHtml = "Strategic Plan 2006-2012";
row.Cells.Add(cell);
table.Rows.Add(row);

// Create the next header row and add it to the table;
row = new HtmlTableRow();
cell = new HtmlTableCell("th");
row.Cells.Add(cell);
cell = new HtmlTableCell();
cell.ColSpan = timeCols + 1;
cell.InnerHtml = " ";
row.Cells.Add(cell);
table.Rows.Add(row);

// Create the next header row and add it to the table;
row = new HtmlTableRow();
cell = new HtmlTableCell("th");
row.Cells.Add(cell);
cell = new HtmlTableCell();
cell.InnerHtml = "Author: Hyperion Solutions";
row.Cells.Add(cell);
cell = new HtmlTableCell();

```



```

cell.Align = "left";
cell.ColSpan = timeCols;
row.Cells.Add(cell);
table.Rows.Add(row);

// Create the next header row and add it to the table;
row = new HtmlTableRow();
cell = new HtmlTableCell("th");
row.Cells.Add(cell);
cell = new HtmlTableCell();
cell.InnerHtml = "SIC Code: 4800";
row.Cells.Add(cell);
cell = new HtmlTableCell();
cell.Align = "left";
cell.ColSpan = timeCols;
row.Cells.Add(cell);
table.Rows.Add(row);

// Create the next header row and add it to the table;
row = new HtmlTableRow();
cell = new HtmlTableCell("th");
row.Cells.Add(cell);
cell = new HtmlTableCell();
cell.InnerHtml = "Scenario: Base";
row.Cells.Add(cell);
cell = new HtmlTableCell();
cell.Align = "left";
cell.ColSpan = timeCols;
row.Cells.Add(cell);
table.Rows.Add(row);

// Create the next header row and add it to the table;
row = new HtmlTableRow();
cell = new HtmlTableCell("th");
row.Cells.Add(cell);
cell = new HtmlTableCell();
cell.InnerHtml = "Millions of Dollars";
row.Cells.Add(cell);
cell = new HtmlTableCell();
cell.Align = "left";
cell.ColSpan = timeCols;
row.Cells.Add(cell);
table.Rows.Add(row);

// Create the next header row and add it to the table;
row = new HtmlTableRow();
cell = new HtmlTableCell("th");
row.Cells.Add(cell);
cell = new HtmlTableCell();
cell.ColSpan = timeCols + 1;
cell.InnerHtml = " ";
row.Cells.Add(cell);
table.Rows.Add(row);

// Create the top header row.
row = new HtmlTableRow();

```

```

        // Create the first two header cells and add them to the
current row.
        cell = new HtmlTableCell("th");
        row.Cells.Add(cell);
        cell = new HtmlTableCell("td");
        row.Cells.Add(cell);

        // Create the time cells.
        for (i = 0; i < timeCols; i++)
        {
            System.Data.DataRow timePeriodRow = timePeriodRows[i];

            cell = new HtmlTableCell();
            cell.Align = "center";
            cell.InnerHtml = (string)timePeriodRow["Name"];
            row.Cells.Add(cell);
        }

        // Add the row to the table.
        table.Rows.Add(row);

        // Create the next header row and add it to the table;
        row = new HtmlTableRow();
        cell = new HtmlTableCell("th");
        row.Cells.Add(cell);
        cell = new HtmlTableCell();
        cell.ColSpan = timeCols + 1;
        cell.InnerHtml = " ";
        row.Cells.Add(cell);
        table.Rows.Add(row);

        foreach (string account in accountList)
        {
            // Create the account row.
            row = new HtmlTableRow();

            if (account == "-")
            {
                cell = new HtmlTableCell("th");
                row.Cells.Add(cell);
                cell = new HtmlTableCell();
                row.Cells.Add(cell);
                for (i = 0; i < timeCols; i++)
                {
                    cell = new HtmlTableCell();
                    cell.InnerHtml = "<hr />";
                    row.Cells.Add(cell);
                }
            }
            else if (account == "=")
            {
                cell = new HtmlTableCell("th");
                row.Cells.Add(cell);
                cell = new HtmlTableCell();
                row.Cells.Add(cell);
                for (i = 0; i < timeCols; i++)
                {

```

```

        cell = new HtmlTableCell();
        cell.InnerHtml = "=====";
        row.Cells.Add(cell);
    }
}
else if (account == " ")
{
    cell = new HtmlTableCell("th");
    row.Cells.Add(cell);
    cell = new HtmlTableCell();
    cell.ColSpan = timeCols + 1;
    cell.InnerHtml = " ";
    row.Cells.Add(cell);
}
else
{
    cell = new HtmlTableCell("th");
    cell.Align = "left";
    cell.InnerHtml = account;
    row.Cells.Add(cell);
    cell = new HtmlTableCell();
    cell.Align = "left";
    accountRows = accountTable.Select("ID='" + account +
    "'");

    if (accountRows.Length > 0)
    {
        cell.InnerHtml = (string)accountRows[0]["Name"];
    }
    row.Cells.Add(cell);
    for (int j = 0; j < timeCols; j++)
    {
        cell = new HtmlTableCell();
        cell.Align = "right";

        try
        {
            {
                System.Data.DataRow[] valueRows =
dataCellsOutTable.Select("Account='" + account + "' AND Time='" +
timePeriodRows[j]["ID"] + "'");
                if (valueRows.Length > 0)
                {
                    Double d = Double.Parse((string)valueRows[0]
["Value"]) / 1000000;

                    cell.InnerHtml = d.ToString("N2");
                }
                else
                {
                    cell.InnerHtml = "N/A";
                }
            }
        }
        catch (Exception)
        {
            cell.InnerHtml = "N/A";
        }
        row.Cells.Add(cell);
    }
}
}

```

```

        table.Rows.Add(row);
    }

    // Add the table to the page.
    this.Controls.Add(table);
}
catch (Exception e)
{
    Response.Write("<center><h3>" + e.Message + "</h3></center>");
}

try
{
    if (hsfEntityWebService != null && entity != null && entity !=
"" && entityOpened)
    {
        hsfEntityWebService.ReleaseEntityLock(sessionID, entity);
    }
}
catch (Exception e)
{
}
}
</script>
<script language="JavaScript">
    function gotosite(site)
    {
        if(site != "")
        {
            self.location=site;
        }
    }
</script>
<html xmlns="http://www.w3.org/1999/xhtml" >
<head id="Head1" runat="server">
    <title>Income Statement</title>
    <style type="text/css">
        th { color: white; background-color: darkgray }
        table { font-size: 90%; }
    </style>
</head>
    <body>
</body>
</html>

```