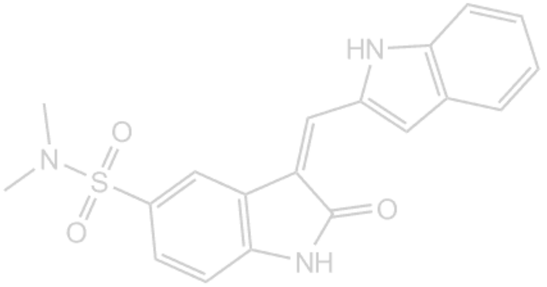
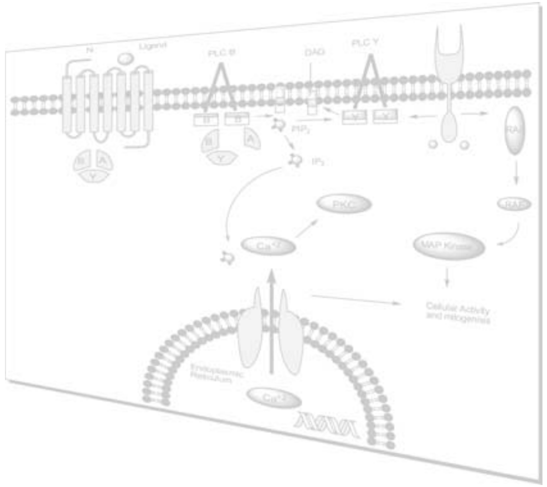
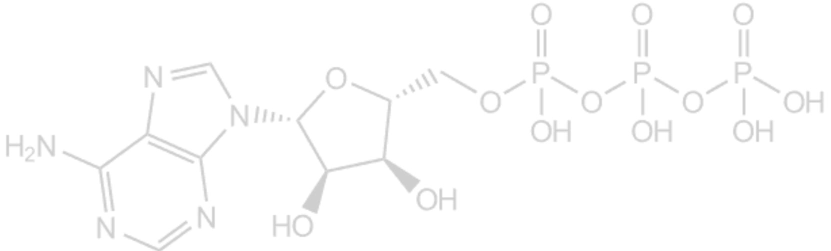
Software Development Kit

Chem & Bio Office Enterprise 2008
Decision Support Platform
Enterprise 10

User and Administrator Guide







CambridgeSoft®
www.cambridgesoft.com

Contents

Preface

About this Guide

Welcome to the Software development Kit. In side this guide, you will find the API(Applied Programming Interface) used by the developers. All the APIs of the Admin Guides are included in this guide that are listed in the “SDK Manual Set”

Each API consists of the:

SYNTAX

INPUTS

OUTPUT

EXAMPLE

This guide is available in print (this file), CHM, and Web-based format.

- Download in CHM format
- Download in PDF format

SDK Manual set

This book includes the API of the following Admin guides:

- BioSar Browser Admin Guide: BioSAR Enterprise is a database management tool, specific to ChemOffice Enterprise. BioSAR Enterprise lets you combine data from multiple applications, such as ChemOffice Enterprise applications and Excel, and subsequently perform searches on the data and display it according to the requirements. The application is fully customizable; allowing you to create you own search forms and display data according to the way you want.
- ChemACX Admin Guide: ChemACX (Available Chemicals Exchange) is a read-only database that contains complete catalogs of the major suppliers of chemicals used in researches and industries. This database is being provided by the chemical manufacturers and distributors. ChemACX is your guide to commercially available chemicals worldwide. By searching the ChemACX database for required chemicals, you can plan a bench synthesis or scale-up a commercial process.
- Inventory Manager Admin Guide: Inventory Enterprise 10.0 is a ChemOffice Enterprise application that allows you to track the data associated with the reagents procured or produced by chemical and pharmaceutical research centers. It keeps track of all the data from the procurement or initial production of the reagents to the depletion or disposal of the reagents.
- Registration System Admin Guide: Registration Enterprise 9.0 is a ChemOffice Enterprise application that allows you to track and characterize the compounds and batches procured or synthesized by the chemical laboratories. It ensures the uniqueness of the compounds by assigning unique IDs to the compounds and providing provisions for automatic duplicate checking.

Inventory Manager API

It Includes the inventory enterprise API:

AddReport

Creates a report layout that can be used to generate reports in the Inventory Enterprise interface. You can add parameters to a report to allow users to filter the report information on the basis of the parameters and view only specific information in the report.

Syntax

```
/cheminv/api/AddReport.asp?ReportName=adVarChar&ReportDisplayName=adVarChar&QueryName=adVarChar&ParamName=adVarChar&ParamDisplayName=adVarChar&ParamType=adVarChar[&QuerySQL=adLongVarChar&ReportDesc=adVarChar&ReportTypeID=adNumeric&NumParams=adNumeric&IsRequired=adNumeric]
```

Inputs

The following heads list the inputs need to be provided for each report parameter:

NOTE: These inputs are not required if the NumParams parameter is set to 0.

REPORTNAME

Data Type: adVarChar

Description: Specifies the name of the MS Access report associated with the report

Options

Required?: Y

Table Name: INV_REPORTS

Database Field Name: REPORTNAME

REPORTDISPLAYNAME

*Data Type:*adVarChar

Description: Specifies the display name of the report. The report display name identifies the report in the Inventory interface.

Options

Required?: Y

Table Name: INV_REPORTS

*Database Field Name:*REPORTDISPLAYNAME

QUERYNAME

*Data Type:*adVarChar

Description: Specifies the MS Access query associated with the report.

Options

Required?: Y

Table Name: INV_REPORTS

Database Field Name: QUERYNAME

QUERYSQL

*Data Type:*adLongVarChar

Description: Specifies the generic SQL query, which does not have parameters associated with it, for the report.

Options

Required?: N

Table Name: INV_REPORTS

Database Field Name: REPORTSQL

REPORTDESC

*Data Type:*adVarChar

Description: Specifies the report description. The report description is displayed above the list of inputs for the report, in the Create Report interface.

Options

Required?: N

Table Name: INV_REPORTS

Database Field Name: REPORT_DESC

REPORTTYPEID

*Data Type:*adNumeric

Description: Specifies the ID associated with the report type.

Options

Required?: N

Table Name: INV_REPORTS

Database Field Name: REPORTTYPE_ID

NUMPARAMS

Data Type:adNumeric

Description: Specifies the number of the report parameters associated with the report.

Options

Required?: N

Table Name: N/A

Database Field Name: N/A

PARAMNAME

*Data Type:*adVarChar

Description: Specifies the database table field associated with the parameter. This field should be specified in the <table-name>.<paramname> format. Reason being, the ParamName field is used in the SQL statement.

Options

Required?: Y

Table Name: INV_REPORTPARAMS

Database Field Name: PARAMNAME

PARAMDISPLAYNAME

*Data Type:*adVarChar

Description: Specifies the display name of the parameter. The parameter display name identifies the parameter in the input interface of the report.

Options

Required?: Y

Table Name: INV_REPORTPARAMS

Database Field Name: PARAMDISPLAYNAME

PARAMTYPE

*Data Type:*adVarChar

Description: Specifies the type of parameter.

*Options: T*ext | Num | Date

Required?: Y

Table Name: INV_REPORTPARAMS

Database Field Name: PARAMTYPE

ISREQUIRED

Data Type: adNumeric

Description: Indicates if the parameter is a required parameter. The IsRequired parameter is set to 1 if the parameter is required, otherwise 0.

Options: 1|*0*

Required?: N

Table Name: INV_REPORTPARAMS

Database Field Name: ISREQUIRED

Output

The ID of the new report, which is stored in the ReportID parameter. The data type of the ReportID parameter is adNumeric.

If the report has some parameters associated with it, then the function also returns the name of the newly created report parameters, which are stored in the ParamName parameter. The data type of the ParamName parameter is adVarChar.

Example

The following code creates a report with display name TestLocationReport. The MS Access report and query associated with the report are rptTestLocation and qryTestLocation, respectively.

```
/cheminv/api/AddReport.asp?ReportName=rptTestLocation&ReportDisplayName=TestLocationReport&QueryName=qryTestLocation&NumParams=0
```

ApproveContainer

Approves or rejects certification of a container, as specified. If a container is approved the status of the container is set to the status defined in the invconfig.ini file for the approved containers, otherwise it is set to the status defined in the invconfig.ini file for the rejected containers.

Syntax

```
/cheminv/api/ApproveContainer.asp?ApprovedContainerIDList=adVarChar&StatusApproved=adNumeric&StatusDefault=adNumeric[&RejectedContainerIDList=adVarChar]
```

Inputs

APPROVEDCONTAINERIDLIST

Data Type: adVarChar

Description: Specifies a comma delimited list of the IDs of containers, whose certification is to be approved.

Options: Y, if RejectedContainerIDList is empty

Required?: N

Table Name: INV_CONTAINERS

Database Field Name: CONTAINER_ID

Output

None.

Example

The following code approves the containers with IDs C1176 and C1004:

```
/cheminv/api/ApproveContainer.asp?ApprovedContainerIDList=C1176,C1004&StatusApproved=1&StatusDefault=1
```

ApproveSampleRequest

Approves or declines a sample request, as specified. If the request is approved, the status of the sample request is set to Approved, otherwise Declined.

Syntax

```
/cheminv/api/ApproveSampleRe-
quest.asp?ApprovedRequestIDList=ad-
VarChar&DeclinedRequestIDList=adVarC
har[&DeclineReasonList=adVarChar]
```

Inputs

APPROVEDREQUESTIDLIST

Data Type: adVarChar

Description: Specifies a comma delimited list of IDs of the requests to be approved.

Options:

Required?: Y, if DeclinedRequestIDList is empty

Table Name: INV_REQUESTS

Database Field Name: REQUEST_ID

DECLINEDREQUESTIDLIST

Data Type: adVarChar

Description: Specifies a comma delimited list of the IDs of the requests to be declined.

Options:

Required?: N

Table Name: INV_REQUESTS

Database Field Name: DECLINE_REASON

Output

None.

Example

The following code approves the sample request with IDs 69, 70, and 43:

```
/cheminv/api/ApproveSampleRe-
quest.asp?ApprovedRequest-
IDList=69,70,43
```

AllotContainer

Samples or splits an existing container into new containers. It also allows you to calculate the amount of substance that is to be left in the container after it is sampled, and specified amount of its contents are transferred to the new containers. The value of the Action parameter specifies whether to sample or split the container.

Syntax

```
/cheminv/api/AllotContainer.asp?Con-
tainerID=adNumeric&NumContainers=ad-
Integer&LocationID=adNumeric&Contain
erTypeID=adNumeric&Action=adVar-
Char&ContainerSize=adNumeric[&QtyRe-
maining=adNumeric&Split#=adNumeric&S
ampleQtyUnit=adNumeric&Barcode-
DescID=adVarChar&DateCertified=adDB-
TimeStamp&ContainerStatusID=adNumeri
c]
```

Inputs

CONTAINERID

Data Type: adNumeric

Description: Specifies the ID of the container to be splitted or sampled.

Options:

Required?: Y

Table Name: INV_CONTAINERS

Database Field Name: CONTAINER_ID

CONTAINERID

Data Type: adNumeric

Description: Specifies the ID of the container to be splitted or sampled.

Options:

Required?: Y

Table Name: INV_CONTAINERS

Database Field Name: CONTAINER_ID

NUMCONTAINERS

Data Type: adInteger

Description: Specifies the number of the new containers to be created after sampling or splitting the container.

Options:

Required?: Y

Table Name: *N/A*

Database Field Name: N/A

LOCATIONID

Data Type: adNumeric

Description: Specifies the location where the newly created containers are to be stored.

Options:

Required?: Y

Table Name: INV_CONTAINERS

Database Field Name: LOCATION_ID_FK

CONTAINERTYPEID

Data Type: adNumeric

Description: Specifies the type of the new containers.

Options:

Required?: Y

Table Name: INV_CONTAINERS

Database Field Name: CONTAINER_TYPE_ID_FK

ACTION

Data Type: adVarChar

Description: Specifies the action to be taken on the container. The Action parameter can take two values, split and sample. If you set the Action parameter to split, all the contents of the original container are transferred to the new containers and the original container is discarded. However, if you set the Action parameter to sample, some of the contents remain there in the container and it is not discarded.

Options: split | sample

Required?: Y

Table Name: N/A

Database Field Name: N/A

CONTAINERSIZE

Data Type: adVarChar

Description: Specifies the size of the containers to be created after sampling or splitting the container.

Options:

Required?: Y

Table Name: INV_CONTAINERS

Database Field Name: QTY_MAX

QTYREMAINING

Data Type: adNumeric

Description: Specifies the quantity of the substance to be left in the original container.

Options:

Required?: if Action = sample

Table Name: INV_CONTAINERS

Database Field Name: QTY_REMAINING

SPLIT#

Data Type: adNumeric

Description: Specifies the quantity of the substance to be allotted to the new containers. A separate Split parameter is associated with each new container. For example, if a container is to be splitted into three containers, the call to AllotContainer should contain three Split parameters, Split1, Split2, and Split3.

Options:

Required?: if Action = split

Table Name: INV_CONTAINERS

Database Field Name: QTY_INITIAL

SAMPLEQTYUNIT

Data Type: adNumeric

Description: Specifies the unit in which the quantity of the contents of the new containers is to be measured.

Options:

Required?: N

Table Name: INV_CONTAINERS

Database Field Name: UNIT_OF_MEAS_ID_FK

BARCODEDESCID

Data Type: adVarChar

Description: Specified the barcode description associated with the new containers.

Options:

Required?: N

Table Name: INV_BARCODE_DESC

Database Field Name: BARCODE_DESC_ID

DATECERTIFIED

Data Type: adDBTimeStamp

Description: Specifies the certification date of the new containers.

Options:

Required?: N

Table Name: INV_CONTAINERS

Database Field Name: DATE_CERTIFIED

CONTAINERSTATUSID

Data Type: adNumeric

Description: Specifies the status for the new containers.

Options:

Required?: *N*

Table Name: INV_CONTAINERS

Database Field Name:

CONTAINER_STATUS_ID_FK

Output

A Pipe (|) delimited list of the IDs of the new containers, which are stored in the ContainerID parameter. For example: 100|101.

Example

The following code splits a container with ID 111 into 4 containers of size 100 and quantity 50:

```
/cheminv/api/AllotContainer.asp?ContainerID=111&NumContainers=4&LocationID=1014&ContainerTypeID=2&Action=split&ContainerSize=100[&split1=50&split2=50&split3=50&split4=50]
```

CalcMolarConc

Calculates the molar concentration of a mixture. The molar concentration of a mixture is calculated according to the number of the moles of the substance and volume and unit of the solvent contained in the mixture.

Syntax

```
/cheminv/api/CalcMolarConc.asp?Mo-
larAmt=adNumeric&SolventVolume=adNu-
meric&SolventVolumeUnitID=adNumeric[
&ConcentrationUnitID=adNumeric&Mo-
larUnitFK=adNumeric]
```

Inputs

MOLARAMOUNT

Data Type: adNumeric

Description: Specifies the number of moles of the substance contained in the mixture.

Options:

Required?: Y

Table Name: INV_PLATES

Database Field Name: MOLAR_AMOUNT

MOLARUNITFK

Data Type: adNumeric

Description: Specifies the ID of the unit in which the molar amount is measured.

Options:

Required?: N

Table Name: INV_PLATES

Database Field Name: MOLAR_UNIT_FK

SOLVENTVOLUME

Data Type: adNumeric

Description: Specifies the volume of solvent contained in the mixture.

Options:

Required?: Y

Table Name: INV_PLATES

Database Field Name: SOLVENT_VOLUME

SOLVENTVOLUMEUNITID

Data Type: adNumeric

Description: Specifies the ID associated with the unit of the volume of the solvent.

Options:

Required?: Y

Table Name: INV_PLATES

Database Field Name: SOLVENT_VOLUME_UNIT_ID_FK

CONCENTRATIONUNITID

Data Type: adNumeric

Description: Specifies the ID of the unit in which the concentration of the mixture is measured.

Options:

Required?: N

Table Name: INV_PLATES

Database Field Name: CONC_UNIT_FK

Output

The molar concentration of the mixture, which is calculated according to the number of the moles of the substance and volume and unit of the solvent contained in the mixture. The molar concentration of the mixture is stored in the MolarConcentration parameter, the data type of which is adNumeric.

Example

The following code allows you to calculate molar concentration of the mixture containing 2 moles of the substance and 100 microliters of the solvent:

```
/cheminv/api/CalcMolarConc.asp?MolarAmt=2&SolventVolume=100&SolventVolumeUnitID=5
```

The SolventVolumeUnitID is set to 5 because the ID of microliters is 5.

CalcVolume

Calculates the volume of the solvent added to a mixture from the concentration of the mixture, and the molar amount and current volume of the solvent in the mixture.

Syntax

```
/cheminv/api/CalcVolumne.asp?MolarAmount=adNumeric&Concentration=adNumeric&ConcentrationUnitID=adNumeric&VolumeUnitID=adNumeric[&CurrSolventVolume=adNumeric&CurrSolventVolumeUnitID=adNumeric]
```

Inputs

MOLARAMOUNT

Data Type: adNumeric

Description: Specifies the molar amount of the solvent.

Options:

Required?: Y

Table Name: INV_PLATES

Database Field Name: MOLAR_AMOUNT

CONCENTRATION

Data Type: adNumeric

Description: Specifies the concentration of the mixture.

Options:

Required?: Y

Table Name: INV_PLATES

Database Field Name: CONCENTRATION

CONCENTRATIONUNITID

Data Type: adNumeric

Description: Specifies the ID of the unit of the concentration.

Options:

Required?: Y

Table Name: INV_PLATES

Database Field Name: CONC_UNIT_FK

VOLUMEUNITID

Data Type: adNumeric

Description: Specifies the ID of the unit in which the volume of the added solvent is to be measured.

Options:

Required?: Y

Table Name: INV_PLATES

Database Field Name: SOLVENT_VOLUME_UNIT_ID_FK

CURRSOLVENTVOLUME

Data Type: adNumeric

Description: Specifies the current volume of solvent.

Options:

Required?: N

Table Name: INV_PLATES

Database Field Name: SOLVENT_VOLUME

CURRSOLVENTVOLUMEUNITID

Data Type: adNumeric

Description: Specifies the ID of the unit in which the current volume of the solvent is measured.

Options:

Required?: N

Table Name: INV_PLATES

Database Field Name: SOLVENT_VOLUME_UNIT_ID_FK

Output

The volume of the added solvent, which is stored in the AddedVolume parameter. The data type of the AddedVolume parameter is adNumeric.

Example

The following code calculates volume of the solvent added to the mixture with concentration 10 and molar amount 2:

```
/cheminv/api/CalcVolumne.asp?MolarAmount=2&Concentration=10&ConcentrationUnitID=1&VolumeUnitID=1
```

CancelRequest

Cancels a request and sets its status as Cancelled.

Syntax

```
/cheminv/api/CancelRequest.asp?RequestID=adNumeric
```

Inputs

REQUESTID

Data Type: adNumeric

Description: Specifies the ID of the request to be cancelled.

Options:

Required?: Y

Table Name: INV_REQUESTS

Database Field Name: REQUEST_ID

Output

The ID of the canceled request, which is stored in the RequestID parameter.

Example

The following code cancels the request with ID 67:

```
/cheminv/api/CancelRequest.asp?RequestID=67
```

CancelOrder

Cancels a container order and sets the order status as Cancelled.

Syntax

```
/cheminv/api/CancelOrder.asp?OrderID=adNumeric[&CancelReason=adVarChar]
```

Inputs

ORDERID

Data Type: adNumeric

Description: Specifies the ID of the order to be canceled.

Options:

Required?: Y

Table Name: INV_ORDERS

Database Field Name: ORDER_ID

CANCELREASON

Data Type: adVarChar

Description: Specifies the reason for canceling the order.

Options:

Required?: N

Table Name: INV_ORDERS

Database Field Name:CANCEL_REASON

Output

The number of orders canceled.

Example

The following code cancels the order with ID 1007:

```
/cheminv/api/CancelOrder.asp?OrderID=1007
```

CertifyContainer

Certifies a container. This API also sets the status, expiration date, and purity of the container, as per the inputs provided by the user.

Syntax

```
/cheminv/api/CertifyContainer.asp?ContainerID=adVarChar&StatusIDFK=adNumeric[&Purity=adNumeric&Interval=adNumeric&StorageConditions=adVarChar&HandlingProcedures=adVarChar]
```

Inputs

CONTAINERID

Data Type: adVarChar

Description: Specifies the ID of the container to be certified.

Options:

Required?: Y

Table Name: INV_CONTAINERS

Database Field Name: CONTAINER_ID

STATUSIDFK

Data Type: adNumeric

Description: Specifies the ID of the status of the container.

Options:

Required?: Y

Table Name: INV_CONTAINERS

Database Field Name: CONTAINER_STATUS_ID_FK

PURITY

Data Type: adNumeric

Description: Specifies the purity of the contents of the container.

Options:

Required?: N

Table Name: INV_CONTAINERS

Database Field Name: PURITY

INTERVAL

Data Type: adNumeric

Description: Specifies the interval after which the container needs to be recertified. The recertification interval is specified in months.

Options:

Required?: N

Table Name: N/A

Database Field Name: N/A

STORAGECONDITIONS

Data Type: adNumeric

Description: Specifies the storage conditions for the container.

Options:

Required?: N

Table Name: INV_CONTAINERS

Database Field Name: STORAGE_CONDITIONS

HANDLINGPROCEDURES

Data Type: adNumeric

Description: Specifies the handling procedures for the container.

Options:

Required?: N

Table Name: INV_CONTAINERS

Database Field Name: HANDLING_PROCEDURES

Output

None.

Example

The following code certifies a container with ID C1002 and sets its status ID to 16:

```
/cheminv/api/CertifyContainer.asp?ContainerID=C1002&StatusIDFK=16
```

CheckDuplicateBarcode

Checks if the barcode that is to be assigned to a container, plate, or location exists already in Inventory Enterprise.

Syntax

```
/cheminv/api/CheckDuplicateBarcode.asp?Barcodes=adVarChar&BarcodeType=adVarChar
```

Inputs

BARCODES

Data Type: adVarChar

Description: Specifies the barcode.

Options:

Required?: Y

Table Name: INV_CONTAINERS,INV_PLATES,INV_LOCATIONS

Database Field Name:

BARCODE,LOCATION_BARCODE,PLATE_BARCODE

BARCODETYPE

Data Type: adVarChar

Description: Specifies the type of the barcode.

Options: Plate|Container|Location

Required?: Y

Table Name: N/A

Database Field Name: N/A

Output

None.

Example

The following code checks whether the container barcode, 295 exists already in Inventory Enterprise:

```
/cheminv/api/CheckDuplicateBar-
code.asp?Barcodes=295&Barcode-
Type=container
```

CheckOutContainer

Checks out a container from its current location for subsequent use at a different location. This API also modifies the current user and status of the container, if specified.

Syntax

```
/cheminv/api/CheckOutContain-
er.asp?LocationID=adNumeric&Contain-
erID=adNumeric&[CurrentUserID=adVarC
har&DefaultLocationID=adNumeric&Own-
erID=adNumeric&ContainerStatusID=ad-
Numeric]
```

Inputs

LOCATIONID

Data Type: adNumeric

Description: Specifies the type of the barcode.

Options:

Required?: Y

Table Name: INV_CONTAINERS

Database Field Name: LOCATION_ID_FK

CONTAINERID

Data Type: adNumeric

Description: Specifies the type of the barcode.

Options:

Required?: Y

Table Name: INV_CONTAINERS

Database Field Name: CONTAINER_ID

CURRENTUSERID

Data Type: adVarChar

Description: Specifies the type of the barcode.

Options:

Required?: N

Table Name: INV_CONTAINERS

Database Field Name: CURRENT_USER_ID_FK

DEFAULTLOCATIONID

Data Type: adNumeric

Description: Specifies the type of the barcode.

Options:

Required?: N

Table Name: INV_CONTAINERS

Database Field Name: LOCATION_ID_FK

OWNERID

Data Type: adNumeric

Description: Specifies the type of the barcode.

Options:

Required?: N

Table Name: INV_CONTAINERS

Database Field Name: OWNER_ID_FK

CONTAINERSTATUSID

Data Type: adNumeric

Description: Specifies the type of the barcode.

Options:

Required?: N

Table Name: INV_CONTAINERS

Database Field Name: CONTAINER_STATUS_ID_FK

Output

The ID of the location where the container is checked out. This ID is stored in the LocationID parameter.

Example

The following code allows you to check out a container with ID 12345 to a different location and change its status to In Use. The IDs of the current location, destination location, and current user of the container are 110, 111, and mlp, respectively.

<PRECLASS="PARAM"

```
/cheminv/api/CheckOutContainer.asp?LocationID=111&ContainerID=12345&[CurrentUserID="mlp"&DefaultLocationID=110&ContainerStatusID="In Use"]
```

CheckSufficientSourcePlateQty

Determines whether or not a plate contains enough content that the requests for the plate contents can be fulfilled. This API assumes that the source plate contains sufficient quantity to fulfill the requests if the amount available in the plate is equal to NumPlates*WellAmt.

Syntax

```
/cheminv/api/CheckSufficientSourcePlateQty.asp?PlateID=adVarChar&WellAmt=adNumeric&NumPlates=adNumeric
```

Inputs

PLATEID

Data Type: adVarChar

Description: Specifies the plate the contents of which are being requested.

Options:

Required?: Y

Table Name: INV_PLATES

Database Field Name: PLATE_ID

WELLAMT

Data Type: adNumeric

Description: Specifies the amount of substance that is to be transferred to each new plate from the source plate.

Options:

Required?: Y

Table Name: N/A

Database Field Name: N/A

NUMPLATES

Data Type: adNumeric

Description: Specifies the total number of plates to which the contents of the source plate are to be transferred.

Options:

Required?: Y

Table Name: N/A

Database Field Name: N/A

Output

Returns 0, 1, 2, or 3, where 0 indicates sufficient quantity, 1 indicates insufficient quantity, 2 indicates remaining quantity for merge, and 3 indicates insufficient quantity in some and remaining in others.

Example

The following code checks whether or not the plate with ID P1001 contains sufficient quantity for fulfilling requests:

```
/cheminv/api/CheckSufficientSourcePlateQty.asp?PlateID=P1001&WellAmt=1&NumPlates=4
```

CloseRequestSample

Closes the specified sample requests and sets their status to Closed.

Syntax

```
/cheminv/api/CloseRequestSample.asp?ClosedRequestIDList=adVarChar
```

Inputs

CLOSEDREQUESTIDLIST

Data Type: adVarChar

Description: Specifies a comma delimited list of the IDs of the sample requests to be closed.

Options:

Required?: Y

Table Name: INV_REQUESTS

Database Field Name: REQUEST_ID

Output

The number of requests closed or the total number of rows updated in the inv_requests table. This number is stored in the NumUpdated parameter, the data type of which is adNumeric.

Example

The following code closes the sample requests with IDs 18 and 20:

```
/cheminv/api/CloseRequestSample.asp?ClosedRequestIDList=18,20
```

CopyContainer

Creates multiple copies of a container. The attributes of the copied containers are exactly similar to the original container.

Syntax

```
/cheminv/api/CopyContainer.asp?ContainerID=adNumeric&NumContainers=adNumeric
```

Inputs

CONTAINERID

Data Type: adNumeric

Description: Specifies the ID of the container to be copied.

Options:

Required?: Y

Table Name: INV_CONTAINERS

Database Field Name: CONTAINER_ID

NUMCONTAINERS

Data Type: adNumeric

Description: Specifies the number of the container copies to be created.

Options:

Required?: Y

Table Name: N/A

Database Field Name: N/A

Output

A Pipe (|) delimited list of the IDs of the new containers. For example: 100|101. These IDs are stored in the NewContainerID parameter, the data type of which is adVarChar.

Example

The following code allows you to create 4 copies of the container with ID 111:

```
/cheminv/api/CopyContainer.asp?ContainerID=111&NumContainers=4
```

CopyDataMap

Creates duplicate copy of a data map.

Syntax

```
/cheminv/api/CopyDataMap.asp?DataMapID=adNumeric[&DataMapName=adVarChar]
```

Inputs

DATAMAPID

Data Type: adNumeric

Description: Specifies the ID of the data map to be copied.

Options:

Required?: Y

Table Name: INV_DATA_MAPS

Database Field Name:DATA_MAP_ID

DATAMAPNAME

Data Type: adVarChar

Description: Specifies the name of the data map.

Options:

Required?: N

Table Name: INV_DATA_MAPS

Database Field Name: DATA_MAP_NAME

Output

The ID of the new data map, which is stored in the DataMapID parameter.

Example

The following code creates copy of the data map with ID 1004:

```
/cheminv/api/CopyDataMap.asp?DataMapID=1004
```

CopyPlate

Creates duplicate copy of a plate.

Syntax

```
/cheminv/api/CopyPlate.asp?PlateID=adNumeric&NumCopies=adNumeric
```

Inputs

PLATEID

Data Type: adNumeric

Description: Specifies the ID of the plate to be copied.

Options:

Required?: Y

Table Name: INV_PLATES

Database Field Name:PLATE_ID

NUMCOPIES

Data Type: adNumeric

Description: Specifies the number of copies to be created.

Options:

Required?: Y

Table Name: N/A

Database Field Name: N/A

Output

The ID of the new plate, which is stored in the NewPlateID parameter. The data type of the NewPlateID parameter is adNumeric.

Example

The following code creates 2 copy of the plate with ID 1004:

```
/cheminv/api/CopyPlate.asp?PlateID=1004&NumCopies=2
```

CreateContainer

Creates a new container and returns its ID.

Syntax

```
/cheminv/api/CreateContainer.asp?LocationID=adNumeric&UOMID=adNumeric&QtyMax=adDouble&QtyInitial=adDouble&ContainerTypeID=adNumeric&ContainerStatusID=adNumeric[&Barcode=adVarChar&BarcodeDescID=adNumeric&CompoundID=adNumeric&RegID=adVarChar&BatchNumber=adNumeric&ExpDate=adDBTimeStamp&MinStockQty=adDouble&MaxStockQty=adDouble&ContainerName=adVarChar&ContainerDesc=adVarChar&TareWeight=adDouble&NetWeight=adDouble&FinalWeight=adDouble&UOWID=adNumeric&Purity=adDouble&UOPID=adNumeric&Concentration=adDouble&Density=adDouble&UOCID=adNumeric&UODID=adNumeric&Grade=adVarChar&SolventIDFK=adVarChar&Comments=adVarChar&StorageConditions=adVarChar&HandlingProcedures=adVarChar&SupplierID=adNumeric&SupplierCatNum=adVarChar&LotNum=adVarChar&DateProduced=adDBTimeStamp&DateOrdered=adDBTimeStamp&DateReceived=adDBTimeStamp&ContainerCost=adNumeric&UOCostID=adNumeric&OwnerID=adVarChar&CurrentUserID=adVarChar&PONumber=adVarChar&POLineNumber=adVarChar&ReqNumber=adVarChar&NumCopies=adNumeric]
```

Inputs

LOCATIONID

Data Type: adNumeric

Description: Specifies the ID of the location where the container is to be stored.

Options:

Required?: Y

Table Name: INV_CONTAINERS

Database Field Name: LOCATION_ID_FK

UOMID

Data Type: adNumeric

Description: Specifies the ID of the unit in which the contents of the container are to be measured. You can determine the ID of an unit using ShowContainerUnits.

Options:

Required?: Y

Table Name: INV_CONTAINERS

Database Field Name: UNIT_OF_MEAS_ID_FK

QTYMAX

Data Type: *adDouble*

Description: Specifies the maximum quantity of the substance that the container can hold.

Options:

Required?: Y

Table Name: INV_CONTAINERS

Database Field Name: QTY_MAX

QTYINITIAL

Data Type: adDouble

Description: Specifies the quantity of the substance to be stored in the container initially.

Options:

Required?: Y

Table Name: INV_CONTAINERS

Database Field Name: QTY_INITIAL

CONTAINERTYPEID

Data Type: adNumeric

Description: Specifies the ID of the type of the container. You can determine the values that ContainerTypeID parameter can take using ShowContainerTypes.

Options:

Required?: Y

Table Name: INV_CONTAINERS

Database Field Name: CONTAINER_TYPE_ID_FK

CONTAINERSTATUSID

Data Type: adNumeric

Description: Specifies the ID of the status of the container. You can determine the values that ContainerStatusID parameter can take using ShowContainerStatus.

Options:

Required?: Y

Table Name: INV_CONTAINERS

Database Field Name: CONTAINER_STATUS_ID_FK

BARCODE

Data Type: adVarChar

Description: Specifies the barcode of the container. The container barcode is an alphanumerical identifier used to identify the container uniquely.

Options:

Required?: N

Table Name: INV_CONTAINERS

Database Field Name: BARCODE

BARCODEDESCID

Data Type: adNumeric

Description: Specifies the ID of the barcode description to be used for the container.

Options:

Required?: N

Table Name: INV_BARCODE_DEC

Database Field Name: BARCODE_DESC_ID

COMPOUNDID

Data Type: adNumeric

Description: Specifies the ID of the substance that is to be stored in the container.

Options:

Required?: N

Table Name: INV_CONTAINERS

Database Field Name: COMPOUND_ID_FK

REGID

Data Type: adVarChar

Description: Specifies the registration ID of the substance that is to be stored in the container.

Options:

Required?: N

Table Name: INV_CONTAINERS

Database Field Name: REG_ID_FK

BATCHNUMBER

Data Type: adNumeric

Description: Specifies the batch number associated with the registration ID of the substance stored in the new container

Options:

Required?: N

Table Name: INV_CONTAINERS

Database Field Name: BATCH_NUMBER_FK

EXPDATE

Data Type: adDBTimeStamp

Description: Specifies the expiration date of the contents of the container.

Options:

Required?: N

Table Name: INV_CONTAINERS

Database Field Name: DATE_EXPIRES

MINSTOCKQTY

Data Type: adDouble

Description: Specifies the minimum stock quantity for the contents of the container. Minimum stock quantity is the minimum quantity of the substance that stock should always hold. Inventory Enterprise issues a re-stock warning if the quantity of the substance in the stock is less than the value of the MinStockQty parameter. The unit for the minimum stock quantity is same as the value of the UOMID parameter.

Options:

Required?: N

Table Name: INV_CONTAINERS

Database Field Name: QTY_MINSTOCK

MAXSTOCKQTY

Data Type: adDouble

Description: Specifies the maximum stock quantity for the contents of the container. Maximum stock quantity is the maximum quantity of the substance that stock can hold. Inventory Enterprise issues a over-stock warning if the quantity of the substance in the stock exceeds than the value of the MaxStockQty parameter. The unit for the maximum stock quantity is same as the value of the UOMID parameter.

Options:

Required?: N

Table Name: INV_CONTAINERS

Database Field Name: QTY_MAXSTOCK

CONTAINERNAME

Data Type: adVarChar

Description: Specifies the name of the container. By default, the ContainerName parameter is set to the internal container ID if no value is specified for this parameter.

Options:

Required?: N

Table Name: INV_CONTAINERS

Database Field Name: CONTAINER_NAME

CONTAINERDESC

Data Type: adVarChar

Description: Specifies the description of the container.

Options:

Required?: N

Table Name: INV_CONTAINERS

Database Field Name: CONTAINER_DESCRIPTION

TAREWEIGHT

Data Type: adDouble

Description:

Specifies the tare weight of the container. The unit for the tare weight of the container is specified in the UOWID parameter.

Options:

Required?: N

Table Name: INV_CONTAINERS

Database Field Name: TARE_WEIGHT

NETWEIGHT

Data Type: adDouble

Description: Specifies the net weight of the container. The unit for the net weight of the container is specified in the UOWID parameter.

Options:

Required?: N

Table Name: INV_CONTAINERS

Database Field Name: NET_WGHT

FINALWEIGHT

Data Type: adDouble

Description: Specifies the final weight of the container. The unit for the final weight of a container is specified in the UOWID parameter.

Options:

Required?: N

Table Name: INV_CONTAINERS

Database Field Name: FINAL_WGHT

UOWID

Data Type: adNumeric

Description: Specifies the ID of the unit in which the tare, net, and final weights of the container are measured. You can determine the values that the UOWID parameter can take using ShowContainerUnits.

Options:

Required?: N

Table Name: INV_CONTAINERS

Database Field Name: UNIT_OF_WGHT_ID_FK

PURITY

Data Type: adDouble

Description: Specifies the purity of the substance specified in the CompoundID parameter.

Options:

Required?: N

Table Name: INV_CONTAINERS

Database Field Name: PURITY

UOPID

Data Type: adNumeric

Description: Specifies the ID of the unit in which the purity of the container contents is measured. You can determine the values that the UOPID parameter can take using ShowContainerUnits.

Options:

Required?: N

Table Name: INV_CONTAINERS

Database Field Name: UNIT_OF_PURITY_ID_FK

CONCENTRATION

Data Type: adDouble

Description: Specifies the concentration of the substance stored in the container.

Options:

Required?: N

Table Name: INV_CONTAINERS

Database Field Name: CONCENTRATION

DENSITY

Data Type: adDouble

Description: Specifies the density of the substance stored in the container.

Options:

Required?: N

Table Name: INV_CONTAINERS

Database Field Name: DENSITY

UOCID

Data Type: adNumeric

Description: Specifies the ID of the unit in which the concentration of the container contents is measured. You can determine the values that the UOCID parameter can take using ShowContainerUnits.

Options:

Required?: N

Table Name: INV_CONTAINERS

Database Field Name: UNIT_OF_COST_ID_FK

UODID

Data Type: adNumeric

Description: Specifies the ID of the unit in which the density of the container contents is measured. You can determine the values that the UODID parameter can take using ShowContainerUnits.

Options:

Required?: N

Table Name: INV_CONTAINERS

Database Field Name: UNIT_OF_DENSITY_ID_FK

GRADE

Data Type: adVarChar

Description:Specifies the grade associated with quality or purity of the container contents.

Options:

Required?: N

Table Name: INV_CONTAINERS

Database Field Name:

GRADE

UODID

Data Type: adNumeric

Description:Specifies the ID of the unit in which the density of the container contents is measured. You can determine the values that the UODID parameter can take using ShowContainerUnits.

Options:

Required?: N

Table Name: INV_CONTAINERS

Database Field Name: UNIT_OF_DENSITY_ID_FK

SOLVENTIDFK

Data Type: adVarChar

Description: Specifies the comments or additional information associated with the container. The Comments parameter can take comments up to 4000 characters. The Comments parameter can be passed as form value, so that it is not truncated if the size of the query string size exceeds the specified limit.

Options:

Required?: N

Table Name: INV_CONTAINERS

Database Field Name: SOLVENT_ID_FK

COMMENTS

Data Type: a*dVarChar*

Description: Specifies the storage conditions for the container.

Options:

Required?: N

Table Name: INV_CONTAINERS

Database Field Name: CONTAINER_COMMENTS

STORAGRECONDITIONS

Data Type: *adVarChar*

Description: Specifies the handling procedures for the container.

Options:

Required?: N

Table Name: INV_CONTAINERS

Database Field Name: STORAGE_CONDITIONS

HANDLINGPROCEDURES

Data Type: *adVarChar*

Description:Specifies the ID of the unit in which the density of the container contents is measured. You can determine the values that the UODID parameter can take using ShowContainerUnits.

Options:

Required?: N

Table Name: INV_CONTAINERS

Database Field Name: HANDLING_PROCEDURES

SUPPLIERID

Data Type: adNumeric

Description:Specifies the ID of the supplier of the container. You can determine the values that the SupplierID parameter can take using ShowSuppliers.

Options:

Required?: N

Table Name: INV_CONTAINERS

Database Field Name: SUPPLIER_ID_FK

SUPPLIERCATNUM

Data Type: adVarChar

Description: Specifies the catalog number that supplier of the container has associated with the container.

Options:

Required?: N

Table Name: INV_CONTAINERS

Database Field Name: SUPPLIER_CATNUM

LOTNUM

Data Type: adVarChar

Description: Specifies the lot number of the supplier of the container.

Options:

Required?: N

Table Name: INV_CONTAINERS

Database Field Name: LOT_NUM

DATEPRODUCED

Data Type: adDBTimeStamp

Description: Specifies the date the contents of the container were produced.

Options:

Required?: N

Table Name: INV_CONTAINERS

Database Field Name: DATE_PRODUCED

DATEORDERED

Data Type: adDBTimeStamp

Description: Specifies the date the container was ordered.

Options:

Required?: N

Table Name: INV_CONTAINERS

Database Field Name: DATE_ORDERED

DATERECIEVED

Data Type: adDBTimeStamp

Description: Specifies the date the container was received.

Options:

Required?: N

Table Name: INV_CONTAINERS

Database Field Name: DATE_RECEIVED

CONTAINERCOST

Data Type: adNumeric

Description: Specifies the cost of the new container.

Options:

Required?: N

Table Name: INV_CONTAINERS

Database Field Name: CONTAINER_COST

UOCOSTID

Data Type: adNumeric

Description: Specifies the unit of the cost of the container. By default, the unit of the cost of the container is USD.

Options:

Required?: N

Table Name: INV_CONTAINERS

Database Field Name: UNIT_OF_COST_ID_FK

OWNERID

Data Type: adVarChar

Description: Specifies the ID of the owner of the container.

Options:

Required?: N

Table Name: INV_CONTAINERS

Database Field Name: OWNER_ID_FK

CURRENTUSERID

Data Type: adVarChar

Description: Specifies the ID of the user of the container.

Options:

Required?: N

Table Name: INV_CONTAINERS

Database Field Name: CURRENT_USER_ID_FK

PONUMBER

Data Type: adVarChar

Description:Specifies the PO number under which the container was ordered.

Options:

Required?: N

Table Name: INV_CONTAINERS

Database Field Name: PO_NUMBER

POLINENUMBER

Data Type: adVarChar

Description: Specifies the PO line number of the container.

Options:

Required?: N

Table Name: INV_CONTAINERS

Database Field Name: PO_LINE_NUMBER

REQNUMBER

Data Type: adVarChar

Description: Specifies the internal requisition number under which the container was ordered.

Options:

Required?: N

Table Name: INV_CONTAINERS

Database Field Name: REQ_NUMBER

NUMCOPIES

Data Type: adNumeric

Description: Specifies the number of copies to be created for the container.

Options:

Required?: N

Table Name: N/A

Database Field Name: N/A

FIELD_1

Data Type: adVarChar

Description: Specifies the value of the configurable field Field_1.

Options:

Required?: N

Table Name: INV_CONTAINERS

Database Field Name: FIELD_1

FIELD_2

Data Type: adVarChar

Description: Specifies the value of the configurable field Field_2.

Options:

Required?: N

Table Name: INV_CONTAINERS

Database Field Name: FIELD_2

FIELD_3

Data Type: adVarChar

Description: Specifies the value of the configurable field Field_3.

Options:

Required?: N

Table Name: INV_CONTAINERS

Database Field Name: FIELD_3

FIELD_4

Data Type: adVarChar

Description: Specifies the value of the configurable field Field_4.

Options:

Required?: N

Table Name: INV_CONTAINERS

Database Field Name: FIELD_4

FIELD_5

Data Type: adVarChar

Description: Specifies the valueof the configurable field Field_5.

Options:

Required?: N

Table Name: INV_CONTAINERS

Database Field Name: FIELD_5

FIELD_6

Data Type: adVarChar

Description: Specifies the value of the configurable field Field_6.

Options:

Required?: N

Table Name: INV_CONTAINERS

Database Field Name: FIELD_6

FIELD_7

Data Type: adVarChar

Description: Specifies the value of the configurable field Field_7.

Options:

Required?: N

Table Name: INV_CONTAINERS

Database Field Name: FIELD_7

FIELD_8

Data Type: adVarChar

Description: Specifies the value of the configurable field Field_8.

Options:

Required?: N

Table Name: INV_CONTAINERS

Database Field Name:FIELD_8

FIELD_9

Data Type: adVarChar

Description: Specifies the value of the configurable field Field_9.

Options:

Required?: N

Table Name: INV_CONTAINERS

Database Field Name: FIELD_9

FIELD_10

Data Type: adVarChar

Description: Specifies the value of the configurable field Field_10.

Options:

Required?: N

Table Name: INV_CONTAINERS

Database Field Name: FIELD_10

DATE_1

Data Type: adDBTimeStamp

Description: Specifies the value of the configurable field Date_1.

Options:

Required?: N

Table Name: INV_CONTAINERS

Database Field Name: DATE_1

DATE_2

Data Type: adDBTimeStamp

Description: Specifies the value of the configurable field Date_2.

Options:

Required?: N

Table Name: INV_CONTAINERS

Database Field Name: DATE_2

DATE_3

Data Type: adDBTimeStamp

Description: Specifies the value of the configurable field Date_3.

Options:

Required?: N

Table Name: INV_CONTAINERS

Database Field Name: DATE_3

DATE_4

Data Type: adDBTimeStamp

Description: Specifies the value of the configurable field Date_4.

Options:

Required?: N

Table Name: INV_CONTAINERS

Database Field Name: DATE_4

DATE_5

Data Type: adDBTimeStamp

Description: Specifies the value of the configurable field Date_5.

Options:

Required?: N

Table Name: INV_CONTAINERS

Database Field Name: DATE_5

Output

A Pipe (|) delimited list of the IDs of the new containers, which are stored in the ContainerID parameter. For example: 100|101. The data type of the ContainerID parameter is adVarChar.

Example

The following code allows you to create a new container with location ID 101, container type ID 13, maximum quantity 200, initial quantity 100, unit of measurement ID 3, and status ID 6:

```
/cheminv/api/CreateContainer.asp?LocationID=101&UOMID=3&QtyMax=200&QtyInitial=100&ContainerTypeID=13&ContainerStatusID=6
```

CreateContainerReport

Causes the RPT report writer to generate a printable Container report. You can use the Container reports to gather information about the attributes of the container or to generate container barcodes.

Syntax

```
/cheminv/api/CreateContainerReport.asp?ContainerList=adVarChar&ReportName=adVarChar[&ReportFormat=adVarChar]
```

Inputs

CONTAINERLIST

Data Type: adVarChar

Description: Specifies a comma delimited list or range of IDs of the containers to be included in the container report. You need to use a Dash (-) to express the range of the container IDs.

Options:

Required?: Y

Table Name: INV_CONTAINERS

Database Field Name: CONTAINER_ID

REPORTNAME

Data Type: adVarChar

Description: Specifies the name of the report template to be used to generate the report. The template name is specified in the similar way as it is defined in the MS Access reports database.

Options:

Required?: Y

Table Name: INV_REPORTS

Database Field Name: REPORTDISPLAYNAME

REPORTFORMAT

Data Type:adVarChar

Description: Specifies the format of the report. A default value for the ReportFormat parameter is defined in the invconfig.ini.

Options: Access Report SnapShot (SNP) | Adobe Acrobat Portable Document Format (PDF) | Rich Text Format (RTF) | Microsoft Excel (XLS)

Required?: N

Table Name:INV_REPORTFORMATS

Database Field Name: FORMATDISPLAYNAME

Output

The file name of the Container report that is saved in the server's reports directory. The location of the server's reports directory is /cfserverasp/RPT/Reports/.

NOTE: To determine the location of the server's reports directory on your server, please check the invconfig.ini file.

Example

The following code generates a container report for container IDs 111, 112, and 113, report name LocationReport2, and report format SNP:

```
/cheminv/api/CreateContainerReport.asp?ContainerList=111,112,113&ReportName=LocationReport2[&ReportFormat=SNP]
```

CreateCustomReport

Causes the RPT report writer to generate a custom report. Custom reports are created by the users, as per their requirements.

Syntax

```
/cheminv/api/CreateCustomReport.asp?ReportName=adVarChar&QueryName=adVarChar&ReportID=adNumeric[&ReportFormat=adVarChar]
```

Inputs

REPORTNAME

Data Type: adVarChar

Description: Specifies the name of the custom report.

Options:

Required?: Y

Table Name: INV_REPORTS

Database Field Name: REPORTDISPLAYNAME

QUERYNAME

Data Type: adVarChar

Description: Specifies the name of the query on the basis of which the custom report is to be created.

Options:

Required?: Y

Table Name: INV_REPORTS

Database Field Name: QUERYNAME

REPORTID

Data Type: adNumeric

Description: Specifies the ID of the report template to be used to generate the report. The template name is specified in the similar way as it is defined in the MS Access reports database.

Options:

Required?: Y

Table Name: INV_REPORTS

Database Field Name: ID

REPORTFORMAT

Data Type: adVarChar

Description: Specifies the format of the report. A default value for the ReportFormat parameter is defined in the invconfig.ini file.

Options: Access Report SnapShot (SNP) | Adobe Acrobat Portable Document Format (PDF) | Rich Text Format (RTF) | Microsoft Excel (XLS)

Required?: N

Table Name: INV_REPORTFORMATS

Database Field Name: FORMATDISPLAYNAME

Output

The file name of the custom report that is saved in the server's reports directory. The location of the server's reports directory is /cfserverasp/RPT/Reports/.

NOTE: To determine the location of the server's reports directory on your server, please check the invconfig.ini file.

Example

The following code creates a custom report with name CustomTestReport:

```
/cheminv/api/CreateCustomReport.asp?ReportName=CustomTestReport&QueryName=qryCustomTest&ReportID=10
```

CreateDataMap

Creates a data map and returns the ID of the newly created data map.

Syntax

```
/cheminv/api/CreateDataMap.asp?DataMapTypeID=adNumeric&DataMapName=adVarChar&DataMapComments=adVarChar&NumColumns=adNumeric&ColumnDelimiter=adVarChar[&NumHeaderRows=adNumeric&DataMapFieldList=adVarChar&DataMapColumnList=adVarChar&UseWellCoordinates=adNumeric]
```

Inputs

DATAMAPTYPEID

Data Type:adNumeric

Description:Specifies the ID of the type of the data map.

Options:

Required?: Y

Table Name: INV_DATA_MAPS

Database Field Name: DATA_MAP_TYPE_ID_FK

DATAMAPNAME

Data Type:adVarChar

Description:Specifies the name of the data map.

Options:

Required?: Y

Table Name: INV_DATA_MAPS

Database Field Name: DATA_MAP_NAME

DATAMAPCOMMENTS

Data Type:adVarChar

Description:Specifies the comments associated with the data map.

Options:

Required?: Y

Table Name: INV_DATA_MAPS

Database Field Name: DATA_MAP_COMMENTS

NUMHEADERROWS

Data Type:adNumeric

Description:Specifies the number of header rows the data map should contain.

Options:

Required?: N

Table Name: INV_DATA_MAPS

Database Field Name: NUM_HEADER_ROWS

NUMCOLUMNS

Data Type:adNumeric

Description:Specifies the number of columns the data map should contain.

Options:

Required?: Y

Table Name: INV_DATA_MAPS

Database Field Name: NUM_COLUMNS

COLUMNDELIMITER

Data Type: adVarChar

Description: Specifies the column delimiter to be used for the data map.

Options:

Required?: Y

Table Name: INV_DATA_MAPS

Database Field Name: COLUMN_DELIMITER

DATAMAPFIELDLIST

Data Type: adVarChar

Description:Specifies the field list for the data map.

Options:

Required?: N

Table Name: N/A

Database Field Name: N/A

DATAMAPCOLUMNLIST

Data Type:adVarChar

Description:Specifies the column list for the data map.

Options:

Required?: N

Table Name: N/A

Database Field Name: N/A

USEWELLCOORDINATES

Data Type: adNumeric

Description:

Options:

Required?: N

Table Name: INV_DATA_MAPS

Database Field Name: USE_WELL_COORDINATES

Output

The ID of the newly created data map, which is stored in the DataMapID parameter. The data type of the DataMapID parameter is adNumeric.

Example

The following code creates a data map, with type ID 19, name TestTemplate, comments creating plates, number of columns 4, and column delimiter tab:

```
/cheminv/api/CreateDataMap.asp?DataMapTypeID=19&DataMapName=TestTemplate&DataMapComments=Creating Plates&NumColumns=4&ColumnDelimiter=tab
```

CreateGridFormat

Creates a grid format and returns the ID of the newly created grid format.

Syntax

```
/cheminv/api/CreateGridFormat.asp?row_count=adNumeric&col_count=adNumeric[&grid_format_type=adNumeric&row_prefix=adVarChar&col_prefix=adVarChar&row_use_letters=adNumeric&col_use_letters=adNumeric&name_separator=adVarChar&number_start_corner=adNumeric]
```

Inputs

GRID_FORMAT_TYPE

*Data Type:*adNumeric

Description:Specifies the type of the grid format.

Options:

Required?: N

Table Name: INV_GRID_FORMAT

Database Field Name: GRID_FORMAT_TYPE_FK

ROW_COUNT

*Data Type:*adNumeric

Description:Specifies the number of rows in the grid.

Options:

Required?: Y

Table Name: INV_GRID_FORMAT

Database Field Name: ROW_COUNT

COL_COUNT

*Data Type:*adNumeric

Description:Specifies the number of columns in the grid.

Options:

Required?: Y

Table Name: INV_GRID_FORMAT

Database Field Name: COL_COUNT

ROW_PREFIX

Data Type: adVarChar

Description: Specifies the prefix to be used for grid rows.

Options:
Required?: N
Table Name: INV_GRID_FORMAT
Database Field Name: ROW_PREFIX

COL_PREFIX
Data Type: adVarChar
Description: Specifies the prefix to be used for grid columns.
Options:
Required?:N
Table Name: INV_GRID_FORMAT
Database Field Name: COL_PREFIX

ROW_USE_LETTERS
Data Type: adNumeric
Description: Specifies whether to label rows with letters or with numbers.
Options:
Required?:N
Table Name: INV_GRID_FORMAT
Database Field Name: ROW_USE_LETTERS

COL_USE_LETTERS
Data Type: adNumeric
Description:Specifies whether to label columns with letters or with numbers.
Options:
Required?: N
Table Name: INV_GRID_FORMAT
Database Field Name: COL_USE_LETTERS

NAME_SEPARATOR
*Data Type:*adVarChar
Description: Specifies name separator to be used in the grid format.
Options:
Required?: N
Table Name: INV_GRID_FORMAT
Database Field Name: NAME_SEPARATOR

NUMBER_START_CORNER
*Data Type:*adNumeric
Description:Specifies start corner for the grid format.
Options:
Required?: N
Table Name: INV_GRID_FORMAT
Database Field Name:
NUMBER_START_CORNER

NUMBER_DIRECTION
*Data Type:*adNumeric
Description:Specifies whether to fill the rows first or to fill the columns first.
Options:
Required?: N
Table Name: INV_GRID_FORMAT
Database Field Name:
NUMBER_DIRECTION

NAME
*Data Type:*adVarChar
Description:Specifies the name of the grid format.
Options:
Required?: Y
Table Name: INV_GRID_FORMAT
Database Field Name: NAME

DESCRIPTION
*Data Type:*adVarChar

Description:Specifies the description of the grid.

Options:

*Required?:*N

Table Name: INV_GRID_FORMAT

Database Field Name: DESCRIPTION

ZERO_PADDING_COUNT

*Data Type:*adNumeric

Description: Specifies zero padding count for the grid format.

Options:

Required?: Y

Table Name: INV_GRID_FORMAT

Database Field Name:

ZERO_PADDING_COUNT

Output

The ID of the new grid format, which is stored in the new_grid_format_id parameter. The data type of the new_grid_format_id parameter is adNumeric.

Example

The following code creates a grid format with row count 8 and column count 12:

```
/cheminv/api/CreateGridFormat.asp?row_count=8&col_count=12
```

CreateLink

Creates a link for a container, substance, or location in Inventory Enterprise. The link is associated with a particular row of an Inventory table that specifies a container, substance, or location.

Syntax

```
/cheminv/api/CreateLink.asp?FK_Value=adVarChar&FK_Name=adVarChar&Table_Name=adVarChar[&URLHref=adVarChar&URLType=adVarChar&ImageSource=adVarChar&LinkText=adVarChar]
```

Inputs

FK_NAME

*Data Type:*adVarChar

Description:Specifies the field that uniquely identifies the row with which the link is to be associated.

Options:

Required?: Y

Table Name: INV_URL

Database Field Name: FK_NAME

FK_VALUE

*Data Type:*adVarChar

Description:Specifies the value of the field specified in the FK_Name parameter.

Options:

Required?: Y

Table Name: INV_URL

Database Field Name: FK_VALUE

TABLE_NAME

*Data Type:*adVarChar

Description:Specifies the table name that contains the row with which the link is to be associated.

Options:

Required?: Y

Table Name: INV_URL

Database Field Name: TABLE_NAME

URLHREF

*Data Type:*adVarChar

Description:Specifies the URL of the link.
Options:

*Required?:*N

Table Name: INV_URL

Database Field Name: URL

URLTYPE

*Data Type:*adVarChar

Description:Specifies the type of the URL.
Options:

*Required?:*N

Table Name: INV_URL

Database Field Name: URL_TYPE

LINKTEXT

*Data Type:*adVarChar

Description: Specifies the text for the link. If no text is specified, the link text is set to the URL of the link.
Options:

*Required?:*N

Table Name: INV_URL

Database Field Name: LINK_TXT

IMAGESOURCE

*Data Type:*adVarChar

Description:Specifies the path of the GIF or JPEG image to be used for the link.
Options:

Required?: N

Table Name: INV_URL

Database Field Name: IMAGE_SRC

Output

The ID of the new link, which is stored in the URL_ID parameter. The data type of the URL_ID parameter is adNumeric.

Example

The following creates link for a container:

```
/cheminv/api/CreateLink.asp?FK_Value=1500&FK_Name=CONTAINER_ID&Table_Name=INV_CONTAINERS
[&URLHref="http://chemfinder.com"&LinkText="TestLink"]
```

CreateLocation

Creates a new location and returns its ID.

Syntax

```
/cheminv/api/CreateLoca-
tion.asp?ParentID=adNumeric&Loca-
tionName=adVarChar(128)[&Barcode=adV
arChar(4000)&LocationDesc=adVar-
Char(255)&LocationTypeID=adNumeric-
GridFormatID=adNumeric&PlateTypeList
=adVarChar&LocationOwnerID=adVar-
Char]
```

Inputs

PARENTID

*Data Type:*adNumeric

Description:Specifies the ID of the parent of the location.
Options:

Required?: Y

Table Name: INV_LOCATIONS
Database Field Name: LOCATION_ID

LOCATIONNAME

*Data Type:*adVarChar
Description:Specifies the name of the location.
Options:
Required?: Y
Table Name: INV_LOCATIONS
Database Field Name: LOCATION_NAME

BARCODEDESCID

*Data Type:*adNumeric
Description:Specifies the ID of the barcode description for the location.
Options:
*Required?:*N
Table Name: INV_BARCODE_DESC
Database Field Name: BARCODE_DESC_ID

BARCODE

*Data Type:*adVarChar
Description:Specifies a unique alphanumerical identifier, which is termed as barcode, for the location.
Options:
*Required?:*N
Table Name: INV_LOCATIONS
Database Field Name: LOCATION_BARCODE

LOCATIONDESC

*Data Type:*adVarChar
Description:Specifies the description of the location.
Options:
Required?: N
Table Name: INV_LOCATIONS
Database Field Name: LOCATION_DESCRIPTION

LOCATIONTYPEID

*Data Type:*adNumeric
Description:Specifies the ID associated with the type of the location. You can determine the values that the LocationTypeID parameter can take using ShowLocationTypes.
Options:
Required?: N
Table Name: INV_LOCATIONS
Database Field Name: LOCATION_TYPE_ID_FK

GRIDFORMATID

*Data Type:*adNumeric
Description:Specifies the ID associated with the grid format of the location. Grid format allows you to create sublocations for the location, while the location is being created.
Options:
Required?: N
Table Name: INV_GRID_FORMAT
Database Field Name: GRID_FORMAT_ID

PLATETYPELIST

*Data Type:*adVarChar
Description:Specifies the plate types that can reside in the location. Commas are used to separate different plate types.
Options:
*Required?:*N
Table Name: N/A
Database Field Name: N/A

LOCATIONOWNERID

*Data Type:*adVarChar

Description:Specifies the ID associated with the user or owner of the location.

Options:

Required?: N

Table Name: INV_LOCATIONS

Database Field Name: OWNER_ID_FK

ALLOWCONTAINERS

*Data Type:*adVarChar

Description:Specifies the list of the containers that are allowed to be stored at the location.

Options:

Required?: N

Table Name: INV_LOCATIONS

Database Field Name: ALLOWS_CONTAINERS

Output

The ID of the new location. The location ID is stored in the LocationID parameter, the data type of which is adNumeric.

Example

The following code creates a location with parent location ID 121 and name Fridge:

```
/cheminv/api/CreateLoca-
tion.asp?ParentID=121&Location-
Name=Fridge
```

CreateLocationReport

Causes the RPT report writer to generate a printable Location report for the specified location. Location report contains information about all the containers or plates stored in the location for which the report is generated. It can be filtered on the basis of the containers, plates, or compounds IDs.

Syntax

```
/cheminv/api/CreateLocationRe-
port.asp?LocationID=adNumeric&Re-
portName=adVarChar&CompoundID=adNume
ric&ContainerID=adNumeric[&Report-
Format=adVarChar]
```

Inputs

LOCATIONID

*Data Type:*adNumeric

Description:Specifies the ID of the location for which the report is to be generated.

Options:

Required?: Y

Table Name: INV_LOCATIONS

*Database Field Name:*LOCATION_ID

REPORTNAME

*Data Type:*adVarChar

Description:Specifies the name of the report template that is to be used to generate the report. The template name is specified in the similar way as it is defined in the MS Access reports database.

Options:

Required?: Y

Table Name: INV_REPORTS

*Database Field Name:*REPORTDISPLAYNAME

COMPOUNDID

*Data Type:*adNumeric

Description:Specifies the compound IDs on the basis of which the containers listed in the Location report are to be filtered.

Options:

Required?: Y

Table Name: INV_COMPOUNDS

*Database Field Name:*COMPOUND_ID

PLATEID OR CONTAINERID

*Data Type:*adNumeric

Description:Specifies the container or plate IDs on the basis of which the containers or plates listed in the Location report are to be filtered.

Options:

Required?: Y

Table Name: INV_CONTAINERS OR INV_PLATES

*Database Field Name:*CONTAINER_ID OR PLATE_ID

REPORTFORMAT

*Data Type:*adVarChar

Description:Specifies the format of the report. A default value for the ReportFormat parameter is defined in the invconfig.ini file.

Options:Access Report SnapShot (SNP) | Adobe Acrobat Portable Document Format (PDF) | Rich Text Format (RTF) | Microsoft Excel (XLS)

*Required?:*N

Table Name:INV_REPORTFORMATS

*Database Field Name:*FORMATDISPLAYNAME

Output

The file name of the Location report that is saved in the server's reports directory. The location of the server's reports directory is /cfserverasp/RPT/Reports/.

NOTE: To determine the location of the server's reports directory on your server, please check the invconfig.ini file.

Example

The following code creates a Location report for location ID 121, template name LReport121, container IDs C1002, C1004, C1025, C1026, compound IDs 11, 19, 182, 125, and report format SNP:

```
/cheminv/api/CreateLocationReport.asp?LocationID=121&ReportName=LReport121&ContainerID=C1002,C1004,C1025,C1026&CompoundID=111,19,182,125[&ReportFormat=SNP
```

CreateOrder

Creates a new order.

Syntax

```
/cheminv/api/CreateOrder.asp?DeliveryLocationID=adNumeric&ShipToName=adVarChar[&ShippingConditions=adVarChar&SampleContainerIDs=adVarChar&StatusID=adNumeric]
```

Inputs

DELIVERYLOCATIONID

*Data Type:*adNumeric

Description:Specifies the ID of the location where the order is to be delivered.

Options:

*Required?:*Y

Table Name:INV_ORDERS

Database Field Name:

DELIVERY_LOCATION_ID_FK

SHIPTONAME

*Data Type:*adVarChar

Description:Specifies the user to whom the order is to be shipped.

Options:

*Required?:*Y

Table Name:INV_ORDERS

*Database Field Name:*SHIP_TO_NAME

SHIPPINGCONDITIONS

*Data Type:*adVarChar

Description:Specifies the shipping conditions for the order.

Options:

*Required?:*N

Table Name:INV_ORDERS

Database Field Name:

SHIPPING_CONDITIONS

SAMPLECONTAINERIDs

*Data Type:*adVarChar

Description: Specifies a comma delimited list of IDs of the containers included in the order.

Options:

*Required?:*N

Table Name:INV_ORDER_CONTAINERS

*Database Field Name:*CONTAINER_ID_FK

STATUSID

*Data Type:*adNumeric

Description:Specifies the ID of the status of the containers included in the order.

Options:

*Required?:*N

Table Name:IINV_CONTAINERS

Database Field Name:

CONTAINER_STATUS_ID_FK

CreatePhysPlateType

Creates a physical plate type and returns the ID of the newly created physical plate type.

Syntax

```
/cheminv/api/CreatePhysPlate-
Type.asp?PhysPlateName=adVar-
Char&RowCount=adNumeric&ColCount=adN
umeric&WellCapacity=Int&Capaci-
tyUnitID=adNumeric&ZeroPadding-
Count=adNumeric[&RowPrefix=adVarChar
&ColPrefix=adVarChar&RowUseLet-
ters=adNumeric&ColUseLetters=adNu-
meric&NameSeparator=adVarChar&Number
StartCorner=adNumeric&NumberDirec-
tion=adNumeric&IsPreBarcoded=adVar-
Char&SupplierID=adNumeric]
```

Inputs

PHYSPLATENAME

*Data Type:*adVarChar

Description:Specifies the name of the physical plate type.

Options:

*Required?:*Y

Table Name:INV_PHYSICAL_PLATE

*Database Field Name:*PHYS_PLATE_NAME

ROWCOUNT

*Data Type:*adNumeric

Description:Specifies the number of rows in the physical plate.

Options:

*Required?:*Y

Table Name:INV_GRID_FORMAT

*Database Field Name:*ROW_COUNT

COLCOUNT

*Data Type:*adNumeric

Description:Specifies the number of columns in the physical plate.

Options:

*Required?:*Y

Table Name:INV_GRID_FORMAT

*Database Field Name:*COL_COUNT

ROWPREFIX

*Data Type:*adVarChar

Description:Specifies the prefix to be used for physical plate rows.

Options:

*Required?:*N

Table Name:INV_GRID_FORMAT

*Database Field Name:*ROW_PREFIX

COLPREFIX

*Data Type:*adVarChar

Description:Specifies the prefix to be used for physical plate columns.

Options:

*Required?:*N

Table Name:INV_GRID_FORMAT

*Database Field Name:*COL_PREFIX

ROWUSEDLETTERS

*Data Type:*adNumeric

Description:Specifies whether to label rows with letters or with numbers. The rows are identified with letters if RowlUseLetters is set to 1.

Options:1 | 0

*Required?:*N

Table Name:INV_GRID_FORMAT

*Database Field Name:*ROW_USE_LETTERS

COLUSEDLETTERS

*Data Type:*adNumeric

Description:Specifies whether to label columns with letters or with numbers. The columns are identified with letters if ColUseLetters is set to 1.

Options:1 | 0

*Required?:*N

Table Name:INV_GRID_FORMAT

*Database Field Name:*COL_USE_LETTERS

NAMESEPARATOR

*Data Type:*adVarChar

Description:Specifies the character that separates the row and column IDs. For example, '-' is the name separator for the following column and row IDs: A-1 and B-1.

Options:

*Required?:*N

Table Name:INV_GRID_FORMAT

*Database Field Name:*NAME_SEPARATOR

NUMBERSTARTCORNER

*Data Type:*adNumeric

Description:Specifies the value with which the plate numbering is to be started.

Options:

*Required?:*N

Table Name:INV_GRID_FORMAT

Database Field Name: NUMBER_START_CORNER

NUMBERDIRECTION

*Data Type:*adNumeric

Description:Specifies whether to fill the rows first or to fill the columns first.

Options:

*Required?:*N

Table Name:INV_GRID_FORMAT

*Database Field Name:*NUMBER_DIRECTION

ISPREBARCODED

*Data Type:*adVarChar

Description:Specifies whether the barcodes should be already defined for the plates created using the new physical plate type. The barcodes are already defined for the plates if IsPreBarcoded is set to Y.

Options: Y | N

Required?: N

Table Name:INV_PHYSICAL_PLATE

*Database Field Name:*IS_PRE_BARCODED

SUPPLIERID

*Data Type:*adNumeric

Description:Specifies the ID of the suppliers of the plates created using the new physical plate type.

Options:

*Required?:*N

Table Name:INV_PHYSICAL_PLATE

*Database Field Name:*SUPPLIER_ID_FK

CAPACITYUNITID

*Data Type:*adNumeric

Description:Specifies the ID associated with the unit in which the well contents are to be measured.

Options:

*Required?:*Y

Table Name:INV_PHYSICAL_PLATE

Database Field Name: CAPACITY_UNIT_ID_FK

WELLCAPACITY

*Data Type:*adNumeric

Description:Specifies the capacity of the wells of the plates created using the new physical plate type.

Options:

*Required?:*Y

Table Name:INV_PHYSICAL_PLATE

*Database Field Name:*WELL_CAPACITY

ZEROPADDINGCOUNT

*Data Type:*adNumeric

Description:Specifies the number of digits with which the column and row numbers are to be padded. Usually, this parameter is set to 2.

Options:

*Required?:*Y

Table Name:INV_GRID_FORMAT

Database Field Name: ZERO_PADDING_COUNT

Output

The ID of the newly created physical plate type. This ID is stored in the NewPhysicalPlateID parameter, the data type of which is adNumeric.

Example

The following code creates a new physical plate type with name NewPlateType, row count 8, column count 12, well capacity 1 ml, well capacity unit ID 9, and supplier ID 1123:

```
/cheminv/api/CreatePhysPlateType.asp?PhysPlateName=NewPlateType&RowCount=8&ColCount=12&WellCapacity=1&CapacityUnitID=9[&SupplierID=1123]
```

CreatePlateFormat

Create a plate format and returns the ID of the newly created plate format.

Syntax

```
/cheminv/api/PlateFormat.asp?PlateFormatName=adVarChar&PhysPlateID=adNumeric
```

Inputs

PLATEFORMATNAME

*Data Type:*adVarChar

Description:Specifies the name of the plate format.

Options:

*Required?:*Y

Table Name:INV_PLATE_FORMAT

Database Field Name:

PLATE_FORMAT_NAME

PHYSPLATEID

*Data Type:*adNumeric

Description:Specifies the ID of the physical plate type associated with the plate format.

Options:

*Required?:*Y

Table Name:INV_PLATE_FORMAT

*Database Field Name:*PHYS_PLATE_ID_FK

Output

The ID of the newly created plate format. The ID is stored in the NewPlateFormatID parameter, the data type of which is adNumeric.

Example

The following code creates a new plate format with name NewPlateFormat and associates physical plate type with ID 111 with it:

```
/cheminv/api/PlateFormat.asp?PlateFormatName=NewPlateFormat&PhysPlateID=111
```

CreatePlateFromMap

Creates plate(s) from a plate map.

Syntax

```
/cheminv/api/CreatePlateFromMap.asp?PlateIDs=adNumeric&NumCopies=adNumeric&ValuePairs=adVarChar&BarcodeDescID=adNumeric
```

Inputs

PLATEIDS

*Data Type:*adNumeric

Description:Specifies the IDs for the new plates.

Options:

*Required?:*Y

Table Name:INV_PLATES

*Database Field Name:*PLATE_ID

NUMCOPIES

*Data Type:*adNumeric

Description:Specifies the number of plate copies to be created.

Options:

*Required?:*Y

Table Name:N/A

*Database Field Name:*N/A

VALUEPAIRS

*Data Type:*adVarChar

Description:Specifies the pairs of the name of the plate fields and the values to be assigned to the fields.

Options:

*Required?:*Y

Table Name:N/A

*Database Field Name:*N/A

BARCODEDESCID

*Data Type:*adNumeric

Description:Specifies the barcode description to be used for the new plates.

Options:

*Required?:*Y

Table Name:INV_BARCODE_DESC

*Database Field Name:*BARCODE_DESC_ID

Output

The ID of the newly created plates, which is stored in the PlateIDs parameter.

Example

The following code creates 2 plates with IDs P1060 and P1061 from the specified plate map:

```
/cheminv/api/CreatePlateFromMap.asp?PlateIDs=P1060,P1061&NumCopies=2&ValuePairs=QTY_INITIAL%3D10::PLATE_TYPE_ID_FK%3D5&BarcodeDescID=1
```

CreatePlateMap

Creates a plate map and returns the ID of the newly created plate map.

Syntax

```
/cheminv/api/CreatePlateMap.asp?LocationID=adNumeric&PlateID=adNumeric
```

Inputs

LOCATIONID

*Data Type:*adNumeric

Description:Specifies the ID of the Plate Map location where the plate map is to be stored.

Options:

*Required?:*Y

Table Name:INV_LOCATIONS

*Database Field Name:*LOCATION_ID

PLATEID

*Data Type:*adNumeric

Description:Specifies the ID of the plate for which the plate map is to be created.

Options:

*Required?:*Y

Table Name:INV_PLATES

*Database Field Name:*PLATE_ID

Output

Example

The following code creates a plate map from a plate with ID A02 and stores the plate map at the location with ID 1032:

```
/cheminv/api/CreatePlateMap.asp?LocationID=1032&PlateID=A02
```

CreatePlateReport

Causes the RPT report writer to generate a printable Plate report. You can use the Plate reports to gather information about the attributes of the plates stored in a particular location.

Syntax

```
/cheminv/api/CreatePlateReport.asp?PlateList=adVarChar&ReportName=adVarChar[&ReportFormat=adVarChar]
```

Inputs

PLATELIST

*Data Type:*adVarChar

Description:Specifies a comma delimited list or range of IDs of the plate to be included in the plate report. You need to use a Dash (-) to express the range of the plate IDs.

Options:

*Required?:*Y

Table Name:INV_PLATES

*Database Field Name:*PLATE_ID

REPORTNAME

*Data Type:*adVarChar

Description:Specifies the name of the report template to be used to generate the report. The template name is specified in the similar way as it is defined in the MS Access reports database.

Options:

*Required?:*Y

Table Name:INV_REPORTS

*Database Field Name:*REPORTDISPLAYNAME

REPORTFORMAT

*Data Type:*adVarChar

Description:Specifies the format of the report. The default value of the ReportFormat parameter is defined in the invconfig.ini file.

Options: Access Report SnapShot (SNP) | Adobe Acrobat Portable Document Format (PDF) | Rich Text Format (RTF) | Microsoft Excel (XLS)

*Required?:*N

Table Name:INV_REPORTFORMATS

*Database Field Name:*FORMATDISPLAYNAME

Output

The file name of the Plate report that is saved in the serverís reports directory. The location of the serverís reports directory is /cfserverasp/RPT/Reports/.

NOTE: To determine the location of the serverís reports directory on your server, please check the invconfig.ini file.

Example

The following code generates a plate report using the TestPlateReport template:

```
/cheminv/api/CreatePlateReport.asp?PlateList=P1000,P1001,P1002,P1003&ReportName=TestPlateReport
```

CreatePlateType

Create a plate type and returns the ID of the newly created plate type.

Syntax

```
/cheminv/api/CreatePlateType.asp?PlateTypeName=adVarChar&MaxFreezeThaw=adNumeric
```

Inputs

PLATETYPENAME

*Data Type:*adVarChar

Description:Specifies the name of the plate type.

Options:

*Required?:*Y

Table Name:INV_PLATE_TYPES

*Database Field Name:*PLATE_TYPE_NAME

MAXFREEZETHAW

*Data Type:*adNumeric

Description:Specifies the maximum number of freeze thaw cycles allowed for the plate type.

Options:

*Required?:*Y

Table Name:INV_PLATE_TYPES

*Database Field Name:*MAX_FREEZE_THAW

Output

The ID of the new plate type. This ID is stored in the NewPlateTypeID parameter, the data type of which is adNumeric.

Example

The following code creates a new plate type with name NewPlateType and maximum allowed freeze thaw cycles 5:

```
/cheminv/api/CreatePlateType.asp?PlateTypeName=NewPlateType&MaxFreezeThaw=5
```

CreateRequest

Creates a new container request.

Syntax

```
/cheminv/api/CreateRequest.asp?ContainerID=adNumeric&UserID=adVarChar&QtyRequired=adNumeric&LocationID=adNumeric&RequestTypeID=adNumeric&ContainerTypeID=adNumeric&NumContainers=adNumeric&QtyList=adVarChar[&DateRequired=Date&RequestComments=adVarChar&ShipToName=adVarChar&RequestStatusID=adNumeric&ExpenseCenter=adVarChar]
```

Inputs

CONTAINERID

*Data Type:*adNumeric

Description:Specifies the ID associated with the container to be requested.

Options:

*Required?:*Y

Table Name:INV_REQUESTS

*Database Field Name:*CONTAINER_ID_FK

USERID

*Data Type:*adVarChar

Description:Specifies the ID associated with the user who has requested the container.

Options:

*Required?:*Y

Table Name:INV_REQUESTS

*Database Field Name:*USER_ID_FK

QTYREQUIRED

*Data Type:*adNumeric

Description:Specifies the quantity of substance that requested container should contain.

Options:

*Required?:*Y

Table Name:INV_REQUESTS

*Database Field Name:*QTY_REQUIRED

LOCATIONID

*Data Type:*adNumeric

Description:Specifies the ID associated with the location, where the requested container is to be delivered.

Options:

*Required?:*Y

Table Name:INV_REQUESTS

Database Field Name:
DELIVERY_LOCATION_ID_FK

DATEREQUIRED

*Data Type:*Date

Description:Specifies the date on which the requested container is to be delivered. The date is specified in the date-time literal format. For example, 01-27-2006.

Options:

*Required?:*N

Table Name:INV_REQUESTS

*Database Field Name:*DATE_REQUIRED

REQUESTCOMMENTS

*Data Type:*adVarChar

Description:Specifies the comments associated with the container request.

Options:

*Required?:*N

Table Name:INV_REQUESTS

Database Field Name:
REQUEST_COMMENTS

REQUESTTYPEID

*Data Type:*adNumeric

Description:Specifies the type of the request.

Options:

*Required?:*Y

Table Name:INV_REQUESTS

Database Field Name:
REQUEST_TYPE_ID_FK

CONTAINERTYPEID

*Data Type:*adNumeric

Description:Specifies the type of the sample containers.

Options:

*Required?:*Y

Table Name:INV_REQUESTS

Database Field Name:
CONTAINER_TYPE_ID_FK

NUMCONTAINERS

*Data Type:*adNumeric

Description:Specifies the number of samples requested.

Options:

*Required?:*Y

Table Name:INV_REQUESTS

Database Field Name:
NUMBER_CONTAINERS

QTYLIST

*Data Type:*adVarChar

Description:Specifies the list of quantities that each sample container should contain.

Options:

*Required?:*Y

Table Name:INV_REQUESTS

*Database Field Name:*QUANTITY_LIST

SHIPTONAME

*Data Type:*adVarChar

Description:Specifies the user to whom the request is to be shipped.

Options:

*Required?:*N

Table Name:INV_REQUESTS

*Database Field Name:*SHIP_TO_NAME

REQUESTSTATUSID

*Data Type:*adNumeric

Description:Specifies the ID of the status of the request.

Options:

*Required?:*N

Table Name:INV_REQUESTS

Database Field Name:
REQUEST_STATUS_ID_FK

EXPENSECENTER

*Data Type:*adVarChar

Description:

Options:

*Required?:*N

Table Name:INV_REQUESTS

*Database Field Name:*EXPENSE_CENTER

Output

The ID of the new request. This ID is stored in the Request_ID parameter, the data type of which is adNumeric.

Example

The following code places request for a container with user ID mjs, container ID 111, date required November 12, 2006, quantity required 10 ml, and location ID 12345:

```
/cheminv/api/CreateRequest.asp?ContainerID=111&UserID=mjs&QtyRequired=10&LocationID=12345&DateRequired=11-13-2003
```

CreateReservation

Allows a user to reserve a specified quantity of the contents of a container.

Syntax

```
/cheminv/api/CreateReservation.asp?ContainerID=adNumeric&ReservationUserID=adVarChar&QtyReserved=adNumeric[&ReservationTypeID=adNumeric]
```

Inputs

CONTAINERID

*Data Type:*adNumeric

Description:Specifies the ID of the container, the contents of which are to be reserved.

Options:

*Required?:*Y

Table Name: INV_RESERVATIONS

*Database Field Name:*CONTAINER_ID_FK

RESERVATIONUSERID

*Data Type:*adVarChar

Description:Specifies the ID of the user, who places the reservation.

Options:

*Required?:*Y

Table Name:INV_RESERVATIONS

*Database Field Name:*USER_ID_FK

QTYRESERVED

*Data Type:*adNumeric

Description:Specifies the amount of the quantity to be reserved.

Options:

*Required?:*Y

Table Name:INV_RESERVATIONS

*Database Field Name:*QTY_RESERVED

RESERVATIONTYPEID

*Data Type:*adNumeric

Description:Specifies the ID associated with the type of reservation.

Options:

*Required?:*N

Table Name: INV_RESERVATIONS

Database Field Name:

RESERVATION_TYPE_ID_FK

Output

The ID of the newly created reservation. This ID is stored in the NewReservationID parameter, the data type of which is adNumeric.

Example

The following code reserves 10 microliters of the substance stored in a container with ID 12312. The ID of the user who reserves the substance is mjs:

```
/cheminv/api/CreateReservation.asp?ContainerID=12312&ReservationUserID=mjs&QtyReserved=10
```

CreateReformatMap

Creates reformat map and returns the ID of the newly created map.

Syntax

```
/cheminv/api/CreateReformatMap.asp?ReformatMapName=adVarChar&ReformatMapType=adVarChar&SourcePlateFormatID=adNumeric&TargetPlateFormatID=adNumeric[&Position1=&Position2=&Position3=&Position4=]
```

Inputs

REFORMATMAPNAME

*Data Type:*adVarChar

Description:Specifies the name of the reformat map.

Options:

*Required?:*Y

Table Name:INV_XMLDOCS

*Database Field Name:*NAME

REFORMATMAPTYPE

*Data Type:*adVarChar

Description:Specifies the type of the reformat map

Options:

*Required?:*Y

Table Name: N/A

Database Field Name: N/A

SOURCEPLATEFORMATID

Data Type: adNumeric

Description:Specifies the ID of the format of the source plate.

Options:

*Required?:*Y

Table Name: N/A

Database Field Name: N/A

TARGETPLATEFORMATID

*Data Type:*adNumeric

Description:Specifies the ID of the format of the target plate.

Options:

*Required?:*Y

Table Name: N/A

Database Field Name: N/A

POSITION1, POSITION2, POSITION3, POSITION4

Data Type:

Description:Specify the order in which the well positions are to be filled.

Options:

Required?: N

Table Name: N/A

Database Field Name: N/A

Output

Example

The following code creates a reformat map with name TestReformtMap, type Stamped, source plate format ID 1000, and target plate format ID 1002:

```
/cheminv/api/CreateReformatMap.asp?ReformatMapName=TestReformtMap&ReformatMapType=adVarChar&SourcePlateFormatID=1000&TargetPlateFormatID=1002
```

CreateSearchResultsReport

Causes the RPT report writer to generate a printable Search results report. Search results report contains the list of the containers or plates obtained in a particular Inventory Enterprise search. This list of containers or plates is termed as hit list.

NOTE: The Search results report for a particular search can be generated only if the hit list for the search exists on the server. Hit list persists on the server only for the duration the user session of the search persists.

Syntax

```
/cheminv/api/CreateSearchResultsReport.asp?ReportName=adVarChar&HitListID=adNumeric[&ReportFormat=adVarChar]
```

Inputs

REPORTNAME

*Data Type:*adVarChar

Description:Specifies the name of the report template to be used to generate the report. The template name is specified in the similar way as it is defined in the MS Access reports database.

Options:

*Required?:*Y

Table Name:INV_REPORTS

*Database Field Name:*REPORTDISPLAYNAME

HITLISTID

*Data Type:*adNumeric

Description:Specifies the ID of the hit list of the search for which the report is to be generated.

Options:

*Required?:*Y

Table Name:N/A

*Database Field Name:*N/A

REPORTFORMAT

Data Type: adVarChar

Description: Specifies the format of the report. A default value for the ReportFormat parameter is defined in the invconfig.ini file.

Options: Access Report SnapShot (SNP) | Adobe Acrobat Portable Document Format (PDF) | Rich Text Format (RTF) | Microsoft Excel (XLS)

*Required?:*N

Table Name: INV_REPORTFORMATS

*Database Field Name:*FORMATDISPLAYNAME

PLATEID OR CONTAINERID

*Data Type:*adNumeric

Description: Specifies the container or plate IDs on the basis of which the containers or plates listed in the Search results report are to be filtered.

Options:

*Required?:*N

Table Name:INV_CONTAINERS or INV_PLATES

*Database Field Name:*CONTAINER_ID or PLATE_ID

Output

The file name of the Search results report that is saved in the server's reports directory. Typically, the location of the server's reports directory is /cfserverasp/RPT/Reports/.

NOTE: To determine the location of the server's reports directory on your server, please check the invconfig.ini file.

Example

The following code generates a Search results report for hit list ID 7, report name SearchResults2, and report format SNP:

```
/cheminv/api/CreateSearchResultsReport.asp?ReportName=SearchResults2&HitListID=7[&ReportFormat=SNP]
```

CreateSubstance

Creates a new substance and returns the ID of the newly created substance. This API also ensures the uniqueness of the substances in the database.

Syntax

```
/cheminv/api/CreateSubstance.asp?SubstanceName=adVarChar[&Structure=CLOB&CAS=adVarChar&ACX_ID=adVarChar&Density=adNumeric&cLogP=adNumeric&Rotatable_Bonds=adNumeric&Tot_Pol_Surf_Area=adNumeric&HBond_Acceptors=adNumeric&HBond_Donors=adNumeric&Alt_ID_1=adVarChar&Alt_ID_2=adVar-
Char&Alt_ID_3=adVarChar&Alt_ID_4=adVarChar&Alt_ID_5=adVarChar]
```

Inputs

SUBSTANCENAME

*Data Type:*adVarChar

Description:Specifies the name of the new substance.

Options:

*Required?:*Y

Table Name:INV_COMPOUNDS

*Database Field Name:*SUBSTANCE_NAME

STRUCTURE

*Data Type:*CLOB

Description:Specifies the chemical structure of the compound. The Structure parameter stores the chemical structure of the substance as the base64 encoded cdx data. Due to the length of cdx data, it is recommended to pass this data as the form data instead of passing it through the query string.

Options:

Required?: N

Table Name:INV_COMPOUNDS

*Database Field Name:*BASE64_CDX

CAS

*Data Type:*adVarChar

Description:Specifies the Chemical Abstracts Service registry number of the substance.

Options:

*Required?:*N

Table Name:INV_COMPOUNDS

*Database Field Name:*CAS

ACX_ID

*Data Type:*adVarChar

Description:Specifies the ACX ID of the substance.

Options:

*Required?:*N

Table Name:INV_COMPOUNDS

*Database Field Name:*ACX_ID

DENSITY

*Data Type:*adNumeric

Description:Specifies the density of the substance.

Options:

*Required?:*N

Table Name: INV_COMPOUNDS

*Database Field Name:*DENSITY

CLOGP

*Data Type:*adNumeric

Description:Specifies the value of the Log P, which is a measure of hydrophobicity of a substance.

Options:

*Required?:*N

Table Name:INV_COMPOUNDS

*Database Field Name:*CLOGP

ROTATABLE_BONDS

*Data Type:*adNumeric

Description:Specifies the number of single skeletal or non-terminal bonds in a molecule. The value of the Rotatable_Bonds parameter gives a measure of the molecular flexibility.

Options:

*Required?:*N

Table Name:INV_COMPOUNDS

*Database Field Name:*ROTATABLE_BONDS

TOT_POL_SURF_AREA

*Data Type:*adNumeric

Description:Specifies the total polar surface area.

Options:

*Required?:*N

Table Name:INV_COMPOUNDS

*Database Field Name:*TOT_POL_SURF_AREA

HBOND_ACCEPTORS

*Data Type:*adNumeric

Description:Specifies the number of the hydrogen bond acceptors.

Options:

*Required?:*N

Table Name:INV_COMPOUNDS

*Database Field Name:*HBOND_ACCEPTORS

HBOND_DONORS

*Data Type:*adNumeric

Description:Specifies the number of the hydrogen bond donors.

Options:

*Required?:*N

Table Name: INV_COMPOUNDS

*Database Field Name:*HBOND_DONORS

ALT_ID_1

*Data Type:*adVarChar

Description:Specifies the value of the configurable field ALT_ID_1.

Options:

*Required?:*N

Table Name:INV_COMPOUNDS

*Database Field Name:*ALT_ID_1

ALT_ID_2

*Data Type:*adVarChar

Description:Specifies the value of the configurable field ALT_ID_2.

Options:

*Required?:*N

Table Name:INV_COMPOUNDS

*Database Field Name:*ALT_ID_2

ALT_ID_3

*Data Type:*adVarChar

Description:Specifies the value of the configurable field ALT_ID_3.

Options:

*Required?:*N

Table Name:INV_COMPOUNDS

*Database Field Name:*ALT_ID_3

ALT_ID_4

*Data Type:*adVarChar

Description:Specifies the value of the configurable field ALT_ID_4.

Options:

*Required?:*N

Table Name:INV_COMPOUNDS

*Database Field Name:*ALT_ID_4

ALT_ID_5

*Data Type:*adVarChar

Description:Specifies the value of the configurable field ALT_ID_5.

Options:

*Required?:*N

Table Name:INV_COMPOUNDS

*Database Field Name:*ALT_ID_5

Output

The ID of the newly created substance. This ID is stored in the compound_id parameter, the data type of which is adNumeric.

Example

The following code creates a new substance with the name NewSubstance and ACX ID X1001304-8:

```
/cheminv/api/CreateSub-
stance.asp?SubstanceName=NewSub-
stance[&ACX_ID=X1001304-8]
```

CreateSynonym

Creates a new synonym for a substance.

Syntax

```
/cheminv/api/CreateSynonym.asp?Com-
poundID=adNumeric&SubstanceName=ad-
VarChar
```

Inputs

COMPOUNDID

*Data Type:*adNumeric

Description:Specifies the ID of the substance for which the synonym is to be created.

Options:

*Required?:*Y

Table Name:INV_SYNONYMS

*Database Field Name:*COMPOUND_ID_FK

SUBSTANCENAME

*Data Type:*adVarChar

Description:Specifies the synonym for the substance.

Options:

*Required?:*Y

Table Name:INV_SYNONYMS

*Database Field Name:*SUBSTANCE_NAME

Output

The ID of the newly created synonym. This ID is stored in the Synonym_ID parameter, the data type of which is adNumeric.

Example

The following code allows you to create a synonym, Carbolic Acid, for a substance with ID 23424:

```
/cheminv/api/CreateSynonym.asp?Com-
poundID=23424&SubstanceName="Car-
bolic Acid"
```

CreateTableRow

Creates a new row in a table within the cheminvdb schema.

Syntax

```
/cheminv/api/CreateTableRow.asp?Ta-
bleName=adVarChar&ValuePairs=adVar-
Char
```

Inputs

TABLENAME

*Data Type:*adVarChar

Description:Specifies the name of the table in the schema, for which the row is to be created.

Options:

*Required?:*Y

Table Name:N/A

*Database Field Name:*N/A

VALUEPAIRS

*Data Type:*adVarChar

Description:Specifies a ì::î delimited list of the pairs of the column name and value, for the new row. For example: colnam1=val1::colnam2=val2.

Options:

*Required?:*Y

Table Name:N/A

*Database Field Name:*N/A

Output

None.

Example

The following code allows you to create a new row in the table, Inv_Suppliers:

```
/cheminv/api/CreateTableRow.asp?TableName=Inv_Suppliers&ValuePairs=Supplier_Name=Wyeth::Supplier_Short_Name=WYT::Supplier_Code=WH1432
```

CreateWellContentType

Creates a new well content type or well format.

NOTE: Only those wells that are assigned the Compound well format can contain a compound.

Syntax

```
/cheminv/api/CreateWellContentType.asp?WellFormatName=adVarChar
```

Input

WELLFORMATNAME

*Data Type:*adVarChar

Description:Specifies the name of the well format.

Options:

*Required?:*Y

Table Name:INV_ENUMERATION

*Database Field Name:*ENUM_VALUE

Output

The ID of the new well format. This ID is stored in the newWellFormatID parameter, the data type of which is adVarChar.

Example

The following code creates a new well format with name NewFormat:

```
/cheminv/api/CreateWellContentType.asp?WellFormatName=NewFormat
```

DeleteCompound

Deletes a compound from the cheminvdb schema.

Syntax

```
/cheminv/api/DeleteCompound.asp?CompoundID=adNumeric
```

Input

COMPOUNDID

*Data Type:*adNumeric

Description:Specifies the ID of the compound to be deleted.

Options:

*Required?:*Y

Table Name:INV_COMPOUNDS

*Database Field Name:*COMPOUND_ID

Output

The number of compounds that are deleted. This value is stored in the NumDeleted parameter, the data type of which is adNumeric.

Example

The following code deletes a compound with ID 123:

```
/cheminv/api/DeleteCompound.asp?CompoundID=123
```

DeleteContainer

Deletes a container from Inventory Enterprise.

Syntax

```
/cheminv/api/DeleteContainer.asp?ContainerID=adNumeric
```

Input

CONTAINERID

*Data Type:*adNumeric

Description:Specifies the ID of the container to be deleted.

Options:

*Required?:*Y

Table Name:INV_CONTAINERS

*Database Field Name:*CONTAINER_ID

Output

None.

Example

The following code allows you to delete a container with ID 12737:

```
/cheminv/api/DeleteContainer.asp?ContainerID=12737
```

DeleteDataMap

Deletes the specified data map.

Syntax

```
/cheminv/api/DeleteDataMap.asp?DataMapID=adNumeric
```

Inputs

DATAMAPID

*Data Type:*adNumeric

Description:Specifies the ID of the data map to be deleted.

Options:

*Required?:*Y

Table Name:INV_DATA_MAPS

*Database Field Name:*DATA_MAP_ID

Output

None.

Example

The following code deletes a data map with ID 1006:

```
/cheminv/api/DeleteDataMap.asp?DataMapID=1006
```

DeleteGridFormat

Deletes the specified grid format.

Syntax

```
/cheminv/api/DeleteGridFormat.asp?Grid_format_ID=adNumeric
```

Input

GRID_FORMAT_ID

*Data Type:*adNumeric

Description:Specifies the ID of the grid format to be deleted.

Options:

*Required?:*Y

Table Name:IINV_GRID_FORMAT

*Database Field Name:*GRID_FORMAT_ID

Output

The ID of the deleted grid format, which is stored in the Grid_format_ID parameter.

Example

The following code deletes a grid format with ID 1:

```
/cheminv/api/DeleteGridFormat.asp?Grid_format_ID=1
```

DeleteLink

Deletes the specified link.

Syntax

```
/cheminv/api/DeleteLink.asp?URLID=adNumeric
```

Input

URLID

*Data Type:*adNumeric

Description:Specifies the ID of the link to be deleted.

Options:

*Required?:*Y

Table Name:INV_URL

*Database Field Name:*URL_ID

Output

The ID of the deleted link, which is stored in the URLID parameter.

Example

The following code deletes a link with ID 1:

```
/cheminv/api/DeleteLink.asp?URLID=1
```

DeleteLocation

Deletes a location from Inventory Enterprise. You can delete a location only if it is empty. The location can be emptied by deleting the containers, plates, and sublocations from it using the DeleteContainer, DeletePlate, and DeleteLocation APIs, respectively.

Syntax

```
/cheminv/api/DeleteLocation.asp?LocationID=adNumeric&Recursively=adInteger
```

Inputs

LOCATIONID

*Data Type:*adNumeric

Description:Specifies the ID of the location to be deleted.

Options:

*Required?:*Y

Table Name: INV_LOCATIONS

*Database Field Name:*LOCATION_ID

RECURSIVELY

*Data Type:*adInteger

Description:Specifies whether to delete the location recursively. When you delete a location recursively, in addition to the location, the sublocations of the location are also deleted. The location is deleted recursively if the Recursively parameter is set to 1.

Options:0 | 1

*Required?:*Y

Table Name:N/A

*Database Field Name:*N/A

Output

The ID of the parent location of the deleted location. This ID is stored in the ParentID parameter, the data type of which is adNumeric.

Example

The following code recursively deletes a location with ID 101:

```
/cheminv/api/DeleteLocation.asp?LocationID=101&Recursively=1
```

DeleteLocations

Deletes multiple locations. You can delete a location only if it is empty. The location can be emptied by deleting the containers, plates, and sublocations from it using the DeleteContainer, DeletePlate, or DeleteLocation APIs, respectively.

Syntax

```
/cheminv/api/DeleteLocations.asp?LocationList=adNumeric
```

Input

LOCATIONLIST

*Data Type:*List of adNumeric

Description:Specifies a comma delimited list or the range of the IDs of the locations that are to be deleted. You need to use dash(-) to define the range of the location IDs.

Options:

*Required?:*Y

Table Name: INV_LOCATIONS

*Database Field Name:*LOCATION_ID

Output

None.

Example

The following code deletes two locations with IDs 111 and 112:

```
/cheminv/api/DeleteLocations.asp?LocationList=111,112
```

DeletePhysPlateType

Deletes the specified physical plate type.

Syntax

```
/cheminv/api/DeletePhysPlateType.asp?PhysPlateID=adNumeric
```

Input

PHYSPLATEID

*Data Type:*adNumeric

Description:Specifies the ID of the physical plate type to be deleted.

Options:

*Required?:*Y

Table Name:INV_PHYSICAL_PLATE

*Database Field Name:*PHYS_PLATE_ID

Output

The ID of the deleted physical plate type, which is stored in the PhysPlateId parameter. The data type of the PhysPlateId parameter is adNumeric.

Example

The following code deletes a physical plate type format with ID 1:

```
/cheminv/api/DeletePhysPlateType.asp?PhysPlateID=1
```

DeletePlate

Deletes the specified plate.

Syntax

```
/cheminv/api/DeletePlate.asp?PlateID=adNumeric
```

Inputs

PLATEID

*Data Type:*adNumeric

Description:Specifies the ID of the plate to be deleted.

Options:

*Required?:*Y

Table Name:INV_PLATES

*Database Field Name:*PLATE_ID

Output

The ID of the deleted plate, which is stored in the plate_id parameter. The data type of the plate_id parameter is adNumeric.

Example

The following code deletes a plate with ID 1111:

```
/cheminv/api/DeletePlate.asp?PlateID=1111
```

DeletePlateFormat

Deletes the specified plate format.

Syntax

```
/cheminv/api/DeletePlateFormatID.asp?PlateFormatID=adNumeric
```

Inputs

PLATEFORMATID

*Data Type:*adNumeric

Description:Specifies the ID of the plate format to be deleted.

Options:

*Required?:*Y

Table Name:INV_PLATE_FORMAT

*Database Field Name:*PLATE_FORMAT_ID

Output

The ID of the deleted plate format, which is stored in the PlateFormatID parameter.

Example

The following code deletes a plate format with ID 1.

```
/cheminv/api/DeletePlateFormatID.asp?PlateFormatID=1
```

DeletePlateType

Deletes the specified plate type.

Syntax

```
/cheminv/api/DeletePlateType.asp?PlateTypeID=adNumeric
```

Inputs

PLATETYPEID

*Data Type:*adNumeric

Description:Specifies the ID of the plate type to be deleted.

Options:

*Required?:*Y

Table Name:INV_PLATE_TYPES

*Database Field Name:*PLATE_TYPE_ID

Output

The ID of the deleted plate type, which is stored in the PlateTypeID parameter.

Example

The following code deletes a plate type with ID 1:

```
/cheminv/api/DeletePlateFormatID.asp?PlateTypeID=1
```

DeleteReformatMap

Deletes the specified reformat map.

Syntax

```
/cheminv/api/DeleteReformatMap.asp?XmlDocID=adNumeric
```

Inputs

XMLDOCID

*Data Type:*adNumeric

Description:Specifies the ID of the reformat map to be deleted.

Options:

*Required?:*Y

Table Name:INV_XMLDOCS

*Database Field Name:*XMLDOC_ID

Output

None.

Example

The following code deletes a reformat map with ID 16:

```
/cheminv/api/DeleteReformatMap.asp?XmlDocID=16
```

DeleteReport

Deletes a report along with the parameters associated with it.

Syntax

```
/cheminv/api/DeleteReport.asp?Report_ID=adNumeric
```

Inputs

REPORT_ID

*Data Type:*adNumeric

Description:Specifies the ID of the report to be deleted.

Options:

*Required?:*Y

Table Name:INV_REPORTS

*Database Field Name:*ID

Output

None.

Example

The following code deletes a report with ID 1006:

```
/cheminv/api/DeleteReport.asp?Report_ID=1006
```

DeleteRequest

Deletes the specified request.

Syntax

```
/cheminv/api/DeleteRequest.asp?RequestID=adNumeric
```

Input

REQUESTID

*Data Type:*adNumeric

Description:Specifies the ID of the request to be deleted.

Options:

*Required?:*YY

Table Name:INV_REQUESTS

*Database Field Name:*REQUEST_ID

Output

The ID of the deleted request, which is stored in the RequestID parameter.

Example

The following code deletes a request with ID 101:

```
/cheminv/api/DeleteRequest.asp?RequestID=101
```

DeleteReservation

Deletes a reservation.

Syntax

```
/cheminv/api/DeleteReservation.asp?ReservationID=adNumeric&ContainerID=adNumeric
```

Inputs

RESERVATIONID

*Data Type:*adNumeric

Description:Specifies the ID of the reservation to be deleted.

Options:

*Required?:*Y

Table Name:INV_RESERVATIONS

*Database Field Name:*RESERVATION_ID

CONTAINERID

*Data Type:*adNumeric

Description:Specifies the ID of the container associated with the reservation.

Options:

*Required?:*Y

Table Name:INV_RESERVATIONS

*Database Field Name:*CONTAINER_ID_FK

Output

The ID of the deleted reservation, which is stored in the ReservationID parameter.

Example

The following code deletes a reservation with ID 5 that is associated with a container with ID 23745:

```
/cheminv/api/DeleteReservation.asp?ReservationID=5&ContainerID=23745
```

DeleteSubstance

Deletes a substance.

Syntax

```
/cheminv/api/DeleteSubstance.asp?CompoundID=adNumeric
```

Input

COMPOUNDID

*Data Type:*adNumeric

Description:Specifies the ID of the substance to be deleted.

Options:

*Required?:*Y

Table Name:INV_COMPOUNDS

*Database Field Name:*COMPOUND_ID

Output

None.

Example

The following code deletes a substance with ID 101.

```
/cheminv/api/DeleteSubstance.asp?CompoundID=101
```

DeleteSynonym

Deletes a synonym of a substance.

Syntax

```
/cheminv/api/DeleteSynonym.asp?SynonymID=adNumeric
```

Input

SYNONYMID

*Data Type:*adNumeric

Description:Specifies the ID of the synonym to be deleted.

Options:

*Required?:*Y

Table Name:INV_SYNONYMS

*Database Field Name:*SYNONYM_ID

Output

The ID of the deleted synonym, which is stored in the SynonymID parameter.

Example

The following code deletes a synonym with ID 45:

```
/cheminv/api/DeleteSynonym.asp?SynonymID=45
```

DeleteTableRow

Deletes row(s) of a table in the cheminvdb schema.

Syntax

```
/cheminv/api/DeleteTableRow.asp?TableName=adVarChar&pkColumnName=adVarChar&pkIDs=adNumeric
```

Inputs

PKCOLUMNNAME

*Data Type:*adVarChar

Description:Specifies the name of the primary key column of the table. This column uniquely identifies the row that is to be deleted.

Options:

*Required?:*Y

Table Name:N/A

*Database Field Name:*N/A

TABLENAME

*Data Type:*adVarChar

Description:Specifies the name of the table.

Options:

*Required?:*Y

Table Name:N/A

*Database Field Name:*N/A

PKIDS

*Data Type:*List of adNumeric

Description:Specifies a comma delimited list of the primary key IDs corresponding to the rows to be deleted.

Options:

*Required?:*Y

Table Name:N/A

*Database Field Name:*N/A

Output

None.

Example

The following code deletes rows from the INV_SUPPLIER table:

```
/cheminv/api/DeleteTableRow.asp?TableName=INV_SUPPLIERS&pkColumnName=Supplier_Name&pkIDs=11,12,14
```

DeleteWellContentType

Deletes a well content or well format type.

Syntax

```
/cheminv/api/DeleteWellContentType.asp?WellFormatID=adNumeric
```

Input

WELLFORMATID

*Data Type:*adNumeric

Description:Specifies the ID of the well format to be deleted

Options:

*Required?:*Y

Table Name:INV_ENUMERATION

*Database Field Name:*ENUM_ID

Output

The ID of the deleted well format, which is stored in the WellFormatID parameter.

Example

The following code deletes a well format with ID 3:

```
/cheminv/api/DeleteWellContentType.asp?WellFormatID=3
```

DeliverRequest

Delivers the requested container(s) to the specified location(s). When the container is delivered, its status and location are modified.

Syntax

```
/cheminv/api/DeliverRequest.asp?DeliveredRequestIDList=adVarChar
```

Input

DELIVEREDREQUESTIDLIST

*Data Type:*adVarChar

Description:Specifies a comma delimited list of the IDs of the requests to be delivered.

Options:

*Required?:*Y

Table Name:INV_REQUESTS

*Database Field Name:*REQUEST_ID

Output

The number of requests delivered, which is stored in the rowCount parameter. The data type of the rowCount parameter is adInteger.

Example

The following code delivers the requests with IDs 23, 26, and 27:

```
/cheminv/api/DeliverRequest.asp?DeliveredRequestIDList=23,26,27
```

EditDataMap

Updates a data map, as specified.

Syntax

```
/cheminv/api/EditDataMap.asp?DataMapTypeID=adNumeric&DataMapID=adNumeric&DataMapName=adVarChar&DataMapComments=adVarChar&NumColumns=adInteger&ColumnDelimiter=adVarChar[&NumHeaderRows=adInteger&DataMapFieldList=adVarChar&DataMapColumnList=adVarChar&UseWellCoordinates=adInteger]
```

Inputs

DATAMAPTYPEID

*Data Type:*adNumeric

Description:Specifies the ID of the type of the data map.

Options:

*Required?:*Y

Table Name:INV_DATA_MAPS

Database Field Name:

DATA_MAP_TYPE_ID_FK

DATAMAPID

*Data Type:*adNumeric

Description:Specifies the ID of the data map.

Options:

*Required?:*Y

Table Name:INV_DATA_MAPS

*Database Field Name:*DATA_MAP_ID

DATAMAPNAME

*Data Type:*adVarChar

Description:Specifies the name of the data map.

Options:

*Required?:*Y

Table Name:INV_DATA_MAPS

*Database Field Name:*DATA_MAP_NAME

DATAMAPCOMMENTS

*Data Type:*adVarChar

Description:Specifies the comments associated with the data map.

Options:

*Required?:*Y

Table Name:INV_DATA_MAPS

Database Field Name:

DATA_MAP_COMMENTS

NUMHEADERROWS

*Data Type:*adInteger

Description:Specifies the number of header rows contained in the data map.

Options:

*Required?:*N

Table Name:INV_DATA_MAPS

*Database Field Name:*NUM_HEADER_ROWS

NUMCOLUMNS

*Data Type:*adInteger

Description:Specifies the number of columns contained in the data map.

Options:

*Required?:*Y

Table Name:INV_DATA_MAPS

*Database Field Name:*NUM_COLUMNS

COLUMNDELIMITER

Data Type: adVarChar

Description:Specifies the column delimiter used for the data map.

Options:

*Required?:*Y

Table Name:INV_DATA_MAPS

*Database Field Name:*COLUMN_DELIMITER

DATAMAPFIELDLIST

*Data Type:*adVarChar

Description:Specifies the field list for the data map.

Options:

*Required?:*N

Table Name:N/A

*Database Field Name:*N/A

DATAMAPCOLUMNLIST

Data Type: adVarChar

Description:Specifies the column list for the data map.

Options:

*Required?:*N

Table Name:N/A

*Database Field Name:*N/A

USEWELLCOORDINATES

*Data Type:*adInteger

Options:

*Required?:*N

Table Name: INV_DATA_MAPS

Database Field Name:
USE_WELL_COORDINATES

Output

The ID of the edited data map, which is stored in the DataMapID parameter.

Example

The following code edits data map with ID 20:

```
/cheminv/api/EditDataMap.asp?Data-
MapTypeID=19&DataMapID=20&DataMap-
Name=TestTemplate&DataMapComments=Cr
eating Plates&NumColumns=8&ColumnDe-
limiter=tab
```

EditLocation

Updates attributes of a location.

Syntax

```
/cheminv/api/EditLocation.asp?Loca-
tionID=adNumeric&LocationName=adVar-
Char[&Barcode=adVarChar&LocationDesc
=adVarChar&LocationTypeID=adNumer-
ic&GridFormatID=adNumeric&Plate-
TypeList=adVarChar&LocationOwnerID=a
dVarChar&PropagateAddress=adNumeric]
```

Inputs

LOCATIONID

*Data Type:*adNumeric

Description:Specifies the ID of the location to be updated.

Options:

*Required?:*Y

Table Name:INV_LOCATIONS

*Database Field Name:*LOCATION_ID

LOCATIONNAME

*Data Type:*adVarChar

Description:Specifies the name of the location.

Options:

*Required?:*Y

Table Name:INV_LOCATIONS

*Database Field Name:*LOCATION_NAME

BARCODE

*Data Type:*adVarChar

Description:Specifies the barcode for the location.

Options:

*Required?:*N

Table Name:INV_LOCATIONS

*Database Field Name:*LOCATION_BARCODE

LOCATIONDESC

*Data Type:*adVarChar

Description:Specifies a description for the location.

Options:

*Required?:*N

Table Name:INV_LOCATIONS

Database Field Name:
LOCATION_DESCRIPTION

LOCATIONTYPEID

*Data Type:*adNumeric

Description:Specifies the ID associated with the type of the location.

Options:

*Required?:*N

Table Name:INV_LOCATIONS

Database Field Name:
LOCATION_TYPE_ID_FK

GRIDFORMATID

*Data Type:*adNumeric

Description:Specifies the ID of the grid format associated with the location.

Options:

*Required?:*N

Table Name:INV_GRID_FORMAT

*Database Field Name:*GRID_FORMAT_ID

PLATETYPELIST

*Data Type:*adVarChar

Description:Specifies the list of the plate types allowed to be stored at the location.

Options:

*Required?:*N

Table Name: N/A

*Database Field Name:*N/A

LOCATIONOWNERID

*Data Type:*adVarChar

Description:Specifies the ID for the owner of the location.

Options:

*Required?:*N

Table Name: INV_LOCATIONS

*Database Field Name:*OWNER_ID_FK

PROPAGATEADDRESS

*Data Type:*adNumeric

Description: Indicates whether to propagate the address of the parent location to its child locations, if no address is specified for the children locations explicitly. The address is propagated if the value of this parameter is 1.

Options:1|0

*Required?:*N

Table Name: N/A

Outputs

The ID of the updated location, which is stored in the LocationID parameter. The data type of the LocationID parameter is adVarChar.

Example

The following code edits a location with ID L1022 and name TestLocation:

```
/cheminv/api/EditLocation.asp?LocationID=L1022&LocationName=TestLocation
```

EditOrder

Edits the attributes of an order.

Syntax

```
/cheminv/api/EditOrder.asp?OrderID=adNumeric&DeliveryLocationID=adNumeric&ShipToName=adVarChar[&ShippingConditions=adVarChar&SampleContainerIDs=adVarChar&StatusID=adNumeric]
```

Inputs

ORDERID

*Data Type:*adNumeric

Description: Specifies the ID of the order to be updated.

Options:

*Required?:*Y

Table Name: INV_ORDERS

*Database Field Name:*ORDER_ID

DELIVERYLOCATIONID

*Data Type:*adNumeric

Description:Specifies the ID of the location where the order is to be delivered.

Options:

*Required?:*Y

Table Name: INV_ORDERS

Database Field Name:

DELIVERY_LOCATION_ID_FK

SHIPTONAME

*Data Type:*adVarChar

Description:Specifies the user to whom the order is to be shipped.

Options:

*Required?:*Y

Table Name:INV_ORDERS

*Database Field Name:*SHIP_TO_NAME

SHIPPINGCONDITIONS

*Data Type:*adVarChar

Description:Specifies the shipping conditions for the order.

Options:

*Required?:*N

Table Name:INV_ORDERS

Database Field Name:

SHIPPING_CONDITIONS

SAMPLECONTAINERIDS

*Data Type:*adVarChar

Description: Specifies a comma delimited list of IDs of the containers included in the order.

Options:

*Required?:*N

Table Name:INV_ORDER_CONTAINERS

Database Field Name:

CONTAINER_ID_FK

STATUSID

*Data Type:*adNumeric

Description: Specifies the ID of the status of the containers included in the order.

Options:

*Required?:*N

Table Name: INV_CONTAINERS

Database Field Name:

CONTAINER_STATUS_ID_FK

Output

The ID of the updated order, which is stored in the OrderID parameter.

Example

The following code updates the delivery location and the user to whom the order with ID 21 is to be shipped:

```
/cheminv/api/EditOrder.asp?OrderID=21&DeliveryLocationID=1014&ShipToName=INVCHEMIST
```

EditReport

Updates attributes of a report.

Syntax

```
/cheminv/api/EditReport.asp?ReportName=adVarChar&ReportDisplayName=adVarChar[&ReportID=adNumeric&QueryName=adVarChar&QuerySQL=adLongVarChar&ReportDesc=adVarChar&ReportTypeID&adNumeric&NumParams=adNumeric]
```

Inputs

REPORTID

*Data Type:*adNumeric

Description: Specifies the ID of the report to be updated.

Options:

*Required?:*N

Table Name:INV_REPORTS

*Database Field Name:*ID

REPORTNAME

*Data Type:*adVarChar

Description:Specifies the name of the MS Access report associated with the report.

Options:

*Required?:*Y

Table Name:INV_REPORTS

*Database Field Name:*REPORTNAME

REPORTDISPLAYNAME

*Data Type:*adVarChar

Description:Specifies the display name of the report. The report display name identifies the report in the Inventory interface.

Options:

*Required?:*Y

Table Name:INV_REPORTS

*Database Field Name:*REPORTDISPLAYNAME

QUERYNAME

*Data Type:*adVarChar

Description:Specifies the MS Access query associated with the report.

Options:

*Required?:*N

Table Name:INV_REPORTS

*Database Field Name:*QUERYNAME

QUERYSQL

*Data Type:*adLongVarChar

Description:Specifies the generic SQL query, which does not have parameters associated with it, for the report.

Options:

*Required?:*N

Table Name:INV_REPORTS

*Database Field Name:*REPORTSQL

REPORTDESC

*Data Type:*adVarChar

Description:Specifies the report description. The report description is displayed above the list of inputs for the report, in the Create Report interface.

Options:

*Required?:*N

Table Name:INV_REPORTS

*Database Field Name:*REPORT_DESC

REPORTTYPEID

*Data Type:*adNumeric

Description:Specifies the ID associated with the report type.

Options:

Required?: N

Table Name:INV_REPORTS

*Database Field Name:*REPORTTYPE_ID

NUMPARAMS

*Data Type:*adNumeric

Description:Specifies the number of the report parameters.

Options:

*Required?:*N

Table Name:N/A

*Database Field Name:*N/A

Output

The ID of the updated report, which is stored in the newReportID parameter. The data type of the newReportID parameter is adNumeric.

Example

The following code updates the report with ID 18:

```
/cheminv/api/EditReport.asp?ReportName=rptTestReport&ReportDisplayName=TestReport[&ReportID=18]
```

EmptyTrash

Deletes the contents of the Trash Location.

Syntax

```
/cheminv/api/EmptyTrash.asp?LocationID=adInteger&TrashType=adVarChar
```

Inputs

LOACTIONID

*Data Type:*adInteger

Description:Specifies the ID of the Trash location.

Options:

*Required?:*Y

Table Name:INV_LOCATIONS

*Database Field Name:*LOCATION_ID

TRASHTYPE

*Data Type:*adVarChar

Description:Specifies the type of trash location.

Options:Containers | Plates

*Required?:*Y

Table Name:N/A

*Database Field Name:*N/A

Output

None.

Example

The following code deletes all the plates stored in the Trash location:

```
/cheminv/api/EmptyTrash.asp?LocationID=3&TrashType=Plates
```

ExcludeContainerTypes

Excludes certain types of containers from a given location.

Syntax

```
/cheminv/api/ExcludeContainerTypes.asp?LocationID=adNumeric[&ContainerTypeIDList=adVarChar&AllowContainers=adVarChar]
```

Inputs

LOCATIONID

*Data Type:*adNumeric

Description:Specifies the ID of the location.

Options:

*Required?:*Y

Table Name:INV_LOCATIONS

*Database Field Name:*LOCATION_ID

CONTAINERTYEIDLIST

*Data Type:*adVarChar

Description:Specifies a comma delimited list of IDs of the container types to be excluded from the location.

Options:

*Required?:*N

Table Name:INV_CONTAINERS

Database Field Name:
CONTAINER_TYPE_ID_FK

ALLOWCONTAINERS

*Data Type:*adVarChar

Description:Indicates whether the location can store only locations or containers can also be stored in it. The location can store containers if the AllowContainers parameter is set to 1, otherwise the location can store only locations.

Options:0 | 1

*Required?:*N

Table Name:INV_LOCATIONS

Database Field Name:
ALLOWS_CONTAINERS

Output

None.

Example

The following code excludes container types with ID 6, 9, and 12 from the location with ID 1013:

```
/cheminv/api/ExcludeContainerTypes.asp?LocationID=1013[&ContainerTypeIDList=6,9,12&AllowContainers=1]
```

GetBatchInfo

Provides information for a particular batch or registered compound.

Syntax

```
/cheminv/api/GetBatchInfo.asp?RegID=adVarChar&RegNumber=adVarChar[&BatchNumber=adNumeric]
```

Inputs

REGID

*Data Type:*adNumeric

Description:Specifies the registration ID of the batch, information about which is to be gathered.

Options:

*Required?:*Y, if RegNumber is not provided.

Table Name:N/A

Database Field Name: N/A

REGNUMBER

*Data Type:*adVarChar

Description:Specifies the registration number of the batch.

Options:

*Required?:*Y, if RegID is not provided.

Table Name: N/A

*Database Field Name:*N/A

BATCHNUMBER

*Data Type:*adNumeric

Description:Specifies the batch number. If a value is specified for RegID but not for BatchNumber, BatchNumber is set to 1.

Options:

*Required?:*N

Table Name:N/A

*Database Field Name:*N/A

Outputs

REG_NUMBER

*Data Type:*adVarChar

Description:Specifies the registration number of the compound. For example, AB-000001. notebook_text adVarchar Specifies the notebook text associated with the batch.

NOTEBOOK_PAGE

*Data Type:*adVarchar

Description:Specifies the notebook reference for the batch. scientist_user_id adVarChar Specifies the user id associated with the batch. The table field associated with this parameter is cs_security.people.user_id.

AMOUNT_UNITS

*Data Type:*adVarChar

Description:Specifies the unit in which the amount of the compound is measured. amount adNumeric Specifies the amount of the substance.

REG_ID

*Data Type:*adNumeric

Description:Specifies an integer reference for the registered compound. This integer reference is typically the non-padded number part of the registry number.

BATCH_NUMBER

*Data Type:*adNumeric

Description:Specifies the batch number of the compound.

Example

The following code provides information for the batch with RegNumber AB12345:

```
/cheminv/api/GetBatchInfo.asp?RegNumber=AB12345
```

The following code provides information about the batch with RegID AB12345 and BatchNumber 3:

GetKeyContainerAttributes

Allows you to determine the key attributes of a particular container.

Syntax

```
/cheminv/api/GetKeyContainerAttributes.asp?ContainerID=adVNumeric&ContainerBarcode=adVarChar
```

Inputs

CONTAINERID

*Data Type:*adNumeric

Description:Specifies the ID of the container, whose key attributes are to be determined.

Options:

*Required?:*Y, if ContainerBarcode is not provided.

Table Name:INV_CONTAINERS

*Database Field Name:*CONTAINER_ID

CONTAINERBARCODE

*Data Type:*adVarChar

Description:Specifies the barcode of the container, whose key attributes are to be determined.

Options:

*Required?:*Y, if ContainerID is not provided.

Table Name:INV_CONTAINERS

*Database Field Name:*BARCODE

Outputs

CONTAINERID

*Data Type:*adNumeric

Description:Specifies the unique ID of the container. ContainerBarcode adVarChar Specifies the unique barcode of the container.

CURRENTUSER

*Data Type:*adVarChar

Description:Specifies the current user of the container.

UOM

*Data Type:*adVarChar

Description:Specifies the unit of measurement for the contents of the container.

QTYREMAINING

*Data Type:*adNumeric

Description:Specifies the quantity of the container contents.

PATH

*Data Type:*adVarChar

Description:Specifies the location of the container.

Example

The following code allows you to view the key attributes of a container with ID C111:

```
/cheminv/api/GetKeyContainerAttributes.asp?ContainerID=C111
```

The following code allows you to view the key attributes of a container with barcode C8BGT17:

```
/cheminv/api/GetKeyContainerAttributes.asp?ContainerBarcode=C8BGT17
```

GetLocationAddress

Retrieves the address attributes for a given location.

Syntax

```
/cheminv/api/GetLocationAddress.asp?LocationID=adNumeric
```

Inputs

LOCATIONID

*Data Type:*adNumeric

Description:Specifies the ID of the location, whose address attributes are to be retrieved.

Options:

*Required?:*Y

Table Name:INV_LOCATIONS

*Database Field Name:*LOCATION_ID

Output

ADDRESS_ID

*Data Type:*adNumeric

Description:Specifies the ID of the address of the location.

CONTACT_NAME

*Data Type:*adVarChar

Description:Specifies the contact name.

ADDRESS1

*Data Type:*adVarChar

Description:Specifies the first line of the address.

ADDRESS2

*Data Type:*adVarChar

Description:Specifies the second line of the address.

ADDRESS3

*Data Type:*adVarChar

Description:Specifies the third line of the address.

ADDRESS4

*Data Type:*adVarChar

Description:Specifies the fourth line of the address.

CITY

*Data Type:*adVarChar

Description:Specifies the name of the city.

STATE_ID_FK

*Data Type:*adNumeric

Description:Specifies the ID of the state.

COUNTRY_ID_FK

*Data Type:*adNumeric

Description:Specifies the ID for the country.

ZIP

*Data Type:*adVarChar

Description:Specifies the zip code.

FAX

*Data Type:*adVarChar

Description:Specifies the fax number.

PHONE

*Data Type:*adVarChar

Description:Specifies the phone number.

EMAIL

*Data Type:*adVarChar

Description:Specifies the email address.

STATE_NAME

*Data Type:*adVarChar

Description:Specifies the name of the state.

STATE_ABBREVIATION

*Data Type:*adVarChar

Description:Specifies the abbreviation for the state.

COUNTRY_NAME

*Data Type:*adVarChar

Description:Specifies the name of the country.

Example

The following code retrieves attributes of the location with ID 1013:

```
/cheminv/api/GetLocationAddress.asp?LocationID=1013
```

GetLocationFromBarcode

Allows you to determine the ID and name of a location on specifying its barcode.

Syntax

```
/cheminv/api/GetLocationFromBarcode.asp?LocationBarcode=adVarChar
```

Input

LOCATIONBARCODE

*Data Type:*adVarChar

Description:Specifies the barcode of the location.

Options:

*Required?:*Y

Table Name:INV_LOCATIONS

*Database Field Name:*LOCATION_BARCODE

Outputs

LOCATIONID

*Data Type:*adNumeric

Description:Specifies the ID of the location.

LOCATIONBARCODE

*Data Type:*adVarChar

Description:Specifies the barcode of the location.

LOCATIONNAME

*Data Type:*adVarChar

Description:Specifies the name of the location.

Example

The following code determines the ID and name of a location with barcode HE123:

```
/cheminv/api/GetLocationFromBarcode.asp?LocationBarcode=HE123
```

GetLocationFromID

Allows you to determine the barcode and name of a location on specifying its ID.

Syntax

```
/cheminv/api/GetLocationFromID.asp?LocationID=adNumeric
```

Input

LOCATIONID

*Data Type:*adNumeric

Description:Specifies the ID of the location, whose details are to be determined.

Options:

*Required?:*Y

Table Name:INV_LOCATIONS

*Database Field Name:*LOCATION_ID

Outputs

LOCATIONID

*Data Type:*adNumeric

Description:Specifies the ID of the location.

LOCATIONBARCODE

*Data Type:*adVarChar

Description:Specifies the barcode of the location.

LOCATIONNAME

*Data Type:*adVarChar

Description:Specifies the name of the location.

Example

The following code determines the barcode and name of a location with ID 123:

```
/cheminv/api/GetLocationFromID.asp?LocationID=123
```

MoveContainer

Moves a container to a new location.

Syntax

```
/cheminv/api/MoveContainer.asp?LocationID=adNumeric&ContainerID=adNumeric
```

Inputs

LocationID

*Data Type:*adNumeric

Description:Specifies the ID of the location where the container is to be moved.

Options:

*Required?:*Y

Table Name:INV_LOCATIONS

*Database Field Name:*LOCATION_ID

ContainerID

*Data Type:*adNumeric

Description:Specifies the ID of the container to be moved.

Options:

*Required?:*Y

Table Name:INV_CONTAINERS

*Database Field Name:*CONTAINER_ID

Output

The ID of the new location, which is stored in the LocationID parameter.

Example

The following code moves a container with ID 13423 to a location with ID 111:

```
/cheminv/api/MoveContainer.asp?Loca-
tionID=111&ContainerID=13432
```

MoveLocation

Change the parent location of a location. When the parent location of a location is modified, the location of the contents of the location, such as containers and plates, is also modified.

Syntax

```
/cheminv/api/MoveLocation.asp?Loca-
tionID=adNumeric&ParentID=adNumeric
```

Inputs

LOCATIONID

*Data Type:*adNumeric

Description:Specifies the ID of the location to be moved.

Options:

*Required?:*Y

Table Name:INV_LOCATIONS

*Database Field Name:*LOCATION_ID

PARENTID

*Data Type:*adNumeric

Description:Specifies the ID of the new parent location of the location.

Options:

*Required?:*Y

Table Name:INV_LOCATIONS

*Database Field Name:*PARENT_ID

Output

The ID of the new parent location of the location. This ID is stored in the ParentID parameter, the data type of which is adNumeric.

Example

The following code modifies the parent location of a location with ID 1221 to a location with ID 111:

```
/cheminv/api/MoveLocation.asp?Loca-
tionID=1221&ParentID=111
```

MovePlate

Moves plates from their current location to the specified location.

Syntax

```
/cheminv/api/MovePlate.asp?LocationID=adNumeric&PlateID=adVarChar&DoFillGrid=adVarChar[&Preview=adVarChar]
```

Inputs

Output

The ID of the moved plate. This ID is stored in the plate_id parameter, the data type of which is adNumeric.

Example

The following code moves a plate with ID P1234 to a location with ID 111:

```
/cheminv/api/MovePlate.asp?LocationID=111&PlateID=P1234
```

OrderContainer

Creates a new container with the specified information and places the container in the On Order location. When the order is received, the container is delivered to the location specified in the Location ID field.

Syntax

```
/cheminv/api/OrderContainer.asp?LocationID=adNumeric&UOMID=adNumeric&QtyMax=adDouble&ContainerTypeID=adNumeric&ContainerStatusID=adNumeric[&Barcode=adVarChar&CompoundID=adNumeric&RegID=adVarChar&BatchNumber=adNumeric&ExpDate=adDBTimeStamp&MinStockQty=adDouble&MaxStockQty=adDouble&ContainerName=adVarChar&ContainerDesc=adVarChar&TareWeight=adDouble&UOWID=adNumeric&Purity=adDouble&UOPID=adNumeric&Concentration=adDouble&UOCID=adNumeric&Grade=adVarChar&Solvent=adVarChar&Comments=adVarChar&SupplierID=adNumeric&SupplierCatNum=adVarChar&LotNum=adVarChar&DateProduced=adDBTimeStamp&DateOrdered=adDBTimeStamp&DateReceived=adDBTimeStamp&ContainerCost=adNumeric&UOCostID=adNumeric&OwnerID=adVarChar&CurrentUserID=adVarChar&PONumber=adVarChar&ReqNumber=adVarChar&NumCopies=adNumeric]
```

Inputs

LOCATIONID

*Data Type:*adNumeric

Description:Specifies the ID of the location where the ordered container is to be delivered.

Options:

*Required?:*Y

Table Name:INV_CONTAINER_ORDER

Database Field Name:

DELIVERY_LOCATION_ID_FK

UOMID

*Data Type:*adNumeric

Description:Specifies the ID of the unit in which the contents of the container are to be measured. You can determine the ID of an unit using ShowContainerUnits.

Options:

*Required?*Y

Table Name:INV_CONTAINERS

Database Field Name:

UNIT_OF_MEAS_ID_FK

QTYMAX

*Data Type:*adDouble

Description:Specifies the maximum quantity that the container can hold.

Options:

*Required?*Y

Table Name:INV_CONTAINERS

*Database Field Name:*QTY_MAX

CONTAINERTYPEID

*Data Type:*adNumeric

Description:Specifies the ID of the type of the container. You can determine the values that ContainerTypeID parameter can take using ShowContainerTypes.

Options:

*Required?:*Y

Table Name:INV_CONTAINERS

Database Field Name:

CONTAINER_TYPE_ID_FK

CONTAINERSTATUSID

*Data Type:*adNumeric

Description:Specifies the ID of the status of the container. You can determine the values that ContainerStatusID parameter can take using ShowContainerStatus.

Options:

*Required?:*Y

Table Name:INV_CONTAINERS

Database Field Name:

CONTAINER_STATUS_ID_FK

BARCODE

*Data Type:*adVarChar

Description:Specifies the barcode of the container. The container barcode is an alphanumerical identifier used to identify the container uniquely.

Options:

*Required?*N

Table Name:INV_CONTAINERS

Database Field Name:

BARCODE

COMPOUNDID

*Data Type:*adNumeric

Description:Specifies the ID of the substance to be stored in the container.

Options:

*Required?*N

Table Name: INV_CONTAINERS

*Database Field Name:*COMPOUND_ID_FK

REGID

*Data Type:*adNumeric

Description:Specifies the registration ID of the substance to be stored in the new container.

Options:

*Required?:*N

Table Name:INV_CONTAINERS

*Database Field Name:*REG_ID_FK

BATCHNUMBER

*Data Type:*adNumeric

Description:Specifies the batch number associated with the registration ID of the substance stored in the new container

Options:

*Required?:*N

Table Name:INV_CONTAINERS

*Database Field Name:*BATCH_NUMBER_FK

EXPDATE

*Data Type:*adDBTimeStamp

Description:Specifies the expiration date of the contents of the new container.

Options:

*Required?:*N

Table Name:INV_CONTAINERS

*Database Field Name:*DATE_EXPIRES

MINSTOCKQTY

*Data Type:*adDouble

Description:Specifies the minimum stock quantity for the contents of the new container. Minimum stock quantity is the minimum quantity of the substance that stock should always hold. Inventory Enterprise issues a re-stock warning if the quantity of the substance in the stock is less than the value of the MinStockQty parameter. The unit for the minimum stock quantity is same as the value of the UOMID parameter.

Options:

*Required?:*N

Table Name:INV_CONTAINERS

*Database Field Name:*QTY_MINSTOCK

MAXSTOCKQTY

*Data Type:*adDouble

Description: Specifies the maximum stock quantity for the contents of the new container. Maximum stock quantity is the maximum quantity of the substance that stock can hold. Inventory Enterprise issues a over-stock warning if the quantity of the substance in the stock exceeds than the value of the MaxStockQty parameter. The unit for the maximum stock quantity is same as the value of the UOMID parameter.

Options:

*Required?:*N

Table Name:INV_CONTAINERS

*Database Field Name:*QTY_MAXSTOCK

CONTAINERNAME

*Data Type:*adVarChar

Description:Specifies the name of the container. By default, the ContainerName parameter is set to the internal container ID if no value is specified for this parameter.

Options:

*Required?:*N

Table Name:INV_CONTAINERS

*Database Field Name:*CONTAINER_NAME

CONTAINERDESC

*Data Type:*adVarChar

Description:Specifies the description of the container.

Options:

*Required?:*N

Table Name:INV_CONTAINERS

Database Field Name:
CONTAINER_DESCRIPTION

TAREWEIGHT

Data Type: adDouble

Description:Specifies the tare weight of the container.

Options:

Required?: N

Table Name:INV_CONTAINERS

*Database Field Name:*TARE_WEIGHT

UOWID

*Data Type:*adNumeric

Description:Specifies the ID of the unit in which the tare weight of the container is measured. You can determine the values that the UOWID parameter can take using ShowContainerUnits.

Options:

*Required?:*N

Table Name:INV_CONTAINERS

Database Field Name: UNIT_OF_WGHT_ID_FK

PURITY

*Data Type:*adDouble

Description:Specifies the purity of the substance specified in the CompoundID parameter.

Options:

*Required?:*N

Table Name:INV_CONTAINERS

*Database Field Name:*PURITY

UOPID

*Data Type:*adNumeric

Description:Specifies the ID of the unit in which the purity of the container contents is measured. You can determine the values that the UOPID parameter can take using ShowContainerUnits.

Options:

*Required?:*N

Table Name:INV_CONTAINERS

Database Field Name: UNIT_OF_PURITY_ID_FK

CONCENTRATION

*Data Type:*adDouble

Description: Specifies the concentration of the substance stored in the container.

Options:

*Required?:*N

Table Name:INV_CONTAINERS

*Database Field Name:*CONCENTRATION

UOCID

*Data Type:*adNumeric

Description:Specifies the ID of the unit in which the concentration of the container contents is measured. You can determine the values that the UOCID parameter can take using ShowContainerUnits.

Options:

*Required?:*N

Table Name: INV_CONTAINERS

Database Field Name: UNIT_OF_COST_ID_FK

GRADE

*Data Type:*adVarChar

Description:Specifies the grade associated with quality or purity of the container contents.

Options:

*Required?:*N

Table Name:INV_CONTAINERS

Database Field Name: GRADE

SOLVENTIDFK

*Data Type:*adVarChar

Description:Specifies the solvent for the contents of the container.

Options:

*Required?:*N

Table Name:INV_CONTAINERS

*Database Field Name:*SOLVENT_ID_FK

COMMENTS

*Data Type:*adVarChar

Description:Specifies the comments or additional information associated with the container. The Comments parameter can take comments up to 4000 characters. The Comments parameter can be passed as form value, so that it is not truncated if the size of the query string size exceeds the specified limit.

Options:

*Required?:*N

Table Name:INV_CONTAINERS

Database Field Name:
CONTAINER_COMMENTS

SUPPLIERID

*Data Type:*adNumeric

Description:Specifies the ID of the supplier of the container. You can determine the values that the SupplierID parameter can take using ShowSuppliers.

Options:

*Required?:*N

Table Name:INV_CONTAINERS

*Database Field Name:*SUPPLIER_ID_FK

SUPPLIERCATNUM

*Data Type:*adVarChar

Description:Specifies the catalog number that supplier of the container has associated with the container.

Options:

*Required?:*N

Table Name:INV_CONTAINERS

*Database Field Name:*SUPPLIER_CATNUM

LOTNUM

*Data Type:*adVarChar

Description:Specifies the lot number of the supplier of the container.

Options:

*Required?:*N

Table Name: INV_CONTAINERS

*Database Field Name:*LOT_NUM

DATEPRODUCED

*Data Type:*adDBTimeStamp

Description:Specifies the date the contents of the container were produced.

Options:

*Required?:*N

Table Name:INV_CONTAINERS

*Database Field Name:*DATE_PRODUCED

DATEORDERED

*Data Type:*adDBTimeStamp

Description:Specifies the date the container was ordered.

Options:

*Required?:*N

Table Name:INV_CONTAINERS

*Database Field Name:*DATE_ORDERED

DATERECIEVED

*Data Type:*adDBTimeStamp

Description:Specifies the date the container was received.

Options:

*Required?:*N

Table Name:INV_CONTAINERS

*Database Field Name:*DATE_RECEIVED

CONTAINERCOST

*Data Type:*adNumeric

Description:Specifies the cost of the container.

Options:

*Required?:*N

Table Name:INV_CONTAINERS

*Database Field Name:*CONTAINER_COST

UOCOSTID

*Data Type:*adNumeric

Description:Specifies the ID associated with the unit in which the cost of the container is measured. By default, the unit of the cost of the container is USD.

Options:

*Required?:*N

Table Name:INV_CONTAINERS

*Database Field Name:*UNIT_OF_COST_ID_FK

OWNERID

*Data Type:*adVarChar

Description:Specifies the ID of the owner of the container.

Options:

*Required?:*N

Table Name:INV_CONTAINERS

*Database Field Name:*OWNER_ID_FK

CURRENTUSERID

*Data Type:*adVarChar

Description:Specifies the ID of the user of the container.

Options:

*Required?:*N

Table Name:INV_CONTAINERS

Database Field Name:
CURRENT_USER_ID_FK

PONUMBER

*Data Type:*adVarChar

Description:Specifies the PO number under which the container was ordered.

Options:

*Required?:*N

Table Name:INV_CONTAINERS

*Database Field Name:*PO_NUMBER

REQNUMBER

*Data Type:*adVarChar

Description:Specifies the internal requisition number under which the container was ordered.

Options:

*Required?:*N

Table Name:INV_CONTAINERS

*Database Field Name:*REQ_NUMBER

NUMCOPIES

*Data Type:*adNumeric

Description:Specifies the number of copies to be created for the container.

Options:

*Required?:*N

Table Name:INV_CONTAINERS

*Database Field Name:*N/A

FIELD_1

*Data Type:*adVarChar

Description:Specifies the value of the configurable field Field_1.

Options:

*Required?:*N

Table Name:INV_CONTAINERS

*Database Field Name:*FIELD_1

FIELD_2

Data Type: adVarChar

Description:Specifies the value of the configurable field Field_2.

Options:

*Required?:*N

Table Name:INV_CONTAINERS

*Database Field Name:*FIELD_2

FIELD_3

*Data Type:*adVarChar

Description:Specifies the value of the configurable field Field_3.

Options:

*Required?:*N

Table Name:INV_CONTAINERS

*Database Field Name:*FIELD_3

FIELD_4

*Data Type:*adVarChar

Description:Specifies the value of the configurable field Field_4.

Options:

*Required?:*N

Table Name:INV_CONTAINERS

*Database Field Name:*FIELD_4

FIELD_5

*Data Type:*adVarChar

Description:Specifies the valueof the configurable field Field_5.

Options:

*Required?:*N

Table Name:INV_CONTAINERS

*Database Field Name:*FIELD_5

FILED_6

*Data Type:*adVarChar

Description:Specifies the value of the configurable field Field_6.

Options:

*Required?:*N

Table Name:INV_CONTAINERS

*Database Field Name:*FIELD_6

FIELD_7

*Data Type:*adVarChar

Description:Specifies the value of the configurable field Field_7.

Options:

*Required?:*N

Table Name:INV_CONTAINERS

*Database Field Name:*FIELD_7

FIELD_8

*Data Type:*adVarChar

Description:Specifies the value of the configurable field Field_8.

Options:

*Required?:*N

Table Name:INV_CONTAINERS

*Database Field Name:*FIELD_8

FIELD_9

*Data Type:*adVarChar

Description:Specifies the value of the configurable field

Field_9.

Options:

*Required?:*N

Table Name:INV_CONTAINERS

*Database Field Name:*FIELD_9

FIELD_10

*Data Type:*adVarChar

Description:Specifies the value of the configurable field Field_10.

Options:

*Required?:*N

Table Name:INV_CONTAINERS

*Database Field Name:*FIELD_10

DATE_1

adDBTimeStamp

Specifies the value of the configurable field Date_1.

Options:

*Required?:*N

Table Name:INV_CONTAINERS

*Database Field Name:*DATE_1

DATE_2

*Data Type:*adDBTimeStamp

Description:Specifies the value of the configurable field Date_2.

Options:

*Required?:*N

Table Name:INV_CONTAINERS

*Database Field Name:*DATE_2

DATE_3

*Data Type:*adDBTimeStamp

Description:Specifies the value of the configurable field Date_3.

Options:

*Required?:*N

Table Name:INV_CONTAINERS

*Database Field Name:*DATE_3

DATE_4

*Data Type:*adDBTimeStamp

Description:Specifies the value of the configurable field Date_4.

Options:

*Required?:*N

Table Name:INV_CONTAINERS

*Database Field Name:*DATE_4

DATE_5

*Data Type:*adDBTimeStamp

Description:Specifies the value of the configurable field Date_5.

Options:

*Required?:*N

Table Name:IINV_CONTAINERS

*Database Field Name:*DATE_5

Output

The ID of the newly created container, which is stored in the NewContainerID parameter. The data type of the NewContainerID parameter is adNumeric.

Example

The following code places order for a container and get it delivered at the location with ID 1044:

```
/cheminv/api/OrderContainer.asp?LocationID=1044&UOMID=9&QtyMax=10&ContainerTypeID=9&ContainerStatusID=1[&CompoundID=250&ContainerCost=10]
```

ReceiveOrder

Receives an order and moves the containers that are part of the order to the specified locations. This API also sets the statuses of the containers.

Syntax

```
/cheminv/api/RecieveOrder.asp?OrderID=adNumeric&ReceivedContainerIDs=adVarChar[&StatusID=adNumeric]
```

Inputs

ORDERID

*Data Type:*adNumeric

Description:Specifies the ID of the order to be received.

Options:

*Required?:*Y

Table Name:INV_ORDERS

*Database Field Name:*ORDER_ID

RECEIVEDCONTAINERIDS

*Data Type:*adVarChar

Description:Specifies a comma delimited list of containers that are part of the order.

Options:

*Required?:*Y

Table Name:INV_CONTAINERS

*Database Field Name:*CONTAINER_ID

STATUSID

*Data Type:*adNumeric

Description:Specifies the ID of the status to be used for the containers.

Options:

*Required?:*N

Table Name:INV_CONTAINERS

Database Field Name:

CONTAINER_STATUS_ID_FK

Output

None.

Example

The following code receives the order with ID 21 and moves the C1006 and C1007 containers, which are part of the order, to the specified delivery location:

```
/cheminv/api/RecieveOrder.asp?OrderID=21&ReceivedContainerIDs=C1006,C1007
```

ReconcileContainer

Rectifies the contents of a location by performing the following tasks:

- Move the misplaced containers to the location being rectified. Misplaced containers are those that are not present at the location being rectified, but should be there.
- Move the missing containers to the Missing location. Missing containers are those that are present at the location being rectified, but should not be there.
- Mark the containers as Verified if they are verified to be present at the location being rectified.

Syntax

```
/cheminv/api/ReconcileContainer.asp?LocationID=adNumeric&VerifiedContainerIDList=adVarChar[&AwolContainerIDList=adVarChar&MissingContainerIDList=adVarChar]
```

Inputs

LOCATIONID

*Data Type:*adNumeric

Description:Specifies the ID of the location, the containers stored in which are to be reconciled.

Options:

*Required?:*Y

Table Name:INV_LOCATIONS

*Database Field Name:*LOCATION_ID

VERIFIEDCONTAINERIDLIST

*Data Type:*adVarChar

Description:Specifies a comma delimited list of the IDs of containers verified as being present at the specified location.

Options:

*Required?:*Y - if AwolContainerIDList and Missing ContainerIDList are empty

Table Name:N/A

*Database Field Name:*N/A

AWOLCONTAINERIDLIST

*Data Type:*adVarChar

Description:Specifies a comma delimited list of the IDs of the misplaced containers.

Options:

*Required?:*Y - if VerifiedContainerIDList and Missing ContainerIDList are empty

Table Name:N/A

*Database Field Name:*N/A

CURRENTUSERID

*Data Type:*adVarChar

Description:Specifies the ID of the current user in the container.

Options:

*Required?:*Y - if VerifiedContainerIDList and Missing ContainerIDList are empty

Table Name:INV_CONTAINERS

Database Field Name:

CURRENT_USER_ID_FK

MAKEDEFAULT

*Data Type:*adNumeric

Description:Specifies the default location.

Table Name:N/A

*Database Field Name:*N/A

MISSINGCONTAINERIDLIST

*Data Type:*adVarChar

Description:Specifies a comma delimited list of the IDs of the missing containers.

Options:

*Required?:*Y - if AwolContainerIDList and VerifiedContainerIDList are empty.

Table Name:N/A

*Database Field Name:*N/A

Output

None.

Example

The following code reconciles the contents of the location with ID 101:

```
/cheminv/api/ReconcileCon-
tainer.asp?LocationID=101&Verified-
ContainerIDList=12345,12245[&AwolCon
tainerIDList=12225]
```

RECONCILEPLATE

Rectifies the contents of a location by performing the following tasks:

- Move the misplaced plates to the location being rectified. Misplaced plates are those that are not present at the location being rectified, but should be there.
- Move the missing plates to the Missing location. Missing plates are those that are present at the location being rectified, but should not be there.
- Mark the plates as Verified if they are verified to be present at the location being rectified.

Syntax

```
/cheminv/api/ReconcilePlate.asp?LocationID=adNumeric&VerifiedContainerIDList=adVarChar[&AwolContainerIDList=adVarChar&MissingContainerIDList=adVarChar]
```

Inputs

LOCATIONID

Data Type: adNumeric

Description:Specifies the ID of the location the contents of which are to be rectified.

Options:

*Required?:*Y

Table Name:INV_LOCATIONS

*Database Field Name:*LOCATION_ID

VERIFIEDCONTAINERIDLIST

Data Type: adVarChar

Description:Specifies a comma delimited list of the IDs of the verified plates.

Options:

*Required?:*Y, if AwolContainerIDList and MissingContainerIDList are empty.

Table Name:N/A

*Database Field Name:*N/A

AWOLCONTAINERIDLIST

*Data Type:*adVarChar

Description:Specifies a comma delimited list of the IDs of the misplaced plates.

Options:

*Required?:*Y, if VerifiedContainerIDList and MissingContainerIDList are empty

Table Name:N/A

*Database Field Name:*N/A

MISSINGCONTAINERIDLIST

*Data Type:*adVarChar

Description:Specifies a comma delimited list of the IDs of the missing plates.

Options:

*Required?:*Y, if VerifiedContainerIDList and MissingContainerIDList are empty

Table Name:N/A

*Database Field Name:*N/A

Output

None.

Example

The following code rectifies the contents of the location with ID 1032:

```
/cheminv/api/ReconcilePlate.asp?LocationID=1032&VerifiedContainerIDList=C1167,C1004,C1032,C1062,C1063,C1064&AwolContainerIDList=C1005,C1007&MissingContainerIDList=C1105,C1106,C1007
```

ReformatPlate

Reformats plates to create a single plate from the compounds of the multiple plates.

Syntax

```
/cheminv/api/ReformatPlate.asp?Sour-
cePLateIDList=adVarChar&Reformat-
MapID=adNumeric&BarcodeList=adVarCha
r&LocationID=adNumeric&Plate-
TypeID=adNumeric&Amt=Int&AmtUnit-
ID=adNumeric&SolventID=adNumeric&Sol
ventVolume=Int&SolventVolumeUnit-
ID=adNumeric[&NumTargetPlates=adNu-
meric]
```

Inputs

SOURCEPLATEIDLIST

*Data Type:*adVarChar

Description:Specifies a comma delimited list of the IDs of the plates to be reformatted.

Options:

*Required?:*Y

Table Name:INV_PLATES

*Database Field Name:*PLATE_ID

REFORMATMAPID

*Data Type:*adNumeric

Description:Specifies the ID of the reformat map to be used for reformatting of the plates.

Options:

*Required?:*Y

Table Name:IN/A

*Database Field Name:*N/A

BARCODELIST

*Data Type:*adVarChar

Description:Specifies a list of the barcodes of the target plates.

Options:

*Required?:*Y

Table Name:INV_PLATES

*Database Field Name:*PLATE_BARCODE

LOCATIONID

*Data Type:*adNumeric

Description:Specifies the ID of the location where the target plates are to be stored.

Options:

*Required?:*Y

Table Name:INV_PLATES

*Database Field Name:*LOCATION_ID_FK

PLATETYPEID

*Data Type:*adNumeric

Description:Specifies the ID associated with the type of the target plates.

Options:

*Required?:*Y

Table Name:INV_PLATES

*Database Field Name:*PLATE_TYPE_ID_FK

AMOUNT

*Data Type:*adNumeric

Description:Specifies the amount to be taken from the source plates.

Options:

*Required?:*Y

Table Name:N/A

*Database Field Name:*N/A

AMOUNTUNITID

*Data Type:*adNumeric

Description:Specifies the ID of the unit in which the amount taken from the source plates is to be measured.

Options:

*Required?:*Y

Table Name:N/A

*Database Field Name:*N/A

SOLVENTID

*Data Type:*adNumeric

Description:Specifies the ID of the solvents with which the source plates are to be solvated.

Options:

*Required?:*Y

Table Name:INV_PLATES

*Database Field Name:*SOLVENT_ID_FK

SOLVENTVOLUME

*Data Type:*adNumeric

Description:Specifies the volume of the solvents.

Options:

Required?: Y

Table Name:INV_PLATES

*Database Field Name:*SOLVENT_VOLUME

SOLVENTVOLUMEUNITID

*Data Type:*adNumeric

Description: Specifies the ID of the unit in which the solvent volume is to be measured.

Options:

*Required?:*Y

Table Name:INV_PLATES

Database Field Name:

SOLVENT_VOLUME_UNIT_ID_FK

NUMTARGETPLATES

*Data Type:*adInteger

Description:Specifies the number of the target plates to be created.

Options:

*Required?:*N

Table Name:IN/A

*Database Field Name:*N/A

Output

A Pipe (|) delimited list of the IDs of the newly created plates. For example: 100|101. This list of plate IDs is stored in the PlateIDs parameter, the data type of which is adVarChar.

Example

The following code reformats plate with ID P1001 to create 2 target plates:

```
/cheminv/api/ReformatPlate.asp?SourcePlateIDList=P1001&ReformatMapID=3&BarcodeList=R1002&PlateTypeID=2&Amt=10&AmtUnitID=2&SolventID=4&SolventVolume=1&SolventVolumeUnitID=2&LocationID=101[&NumTargetPlates=2]
```

ReorderContainer

Places order for an already ordered or existing container.

Syntax

```
/cheminv/api/ReorderContainer.asp?DeliveryLocationID=adNumeric&ContainerID=adNumeric&ContainerName=adVarChar&Comments=adVarChar&OwnerID=adVarChar&CurrentUserID=adVarChar&NumCopies=adNumeric&Project=adVarChar&Job=adVarChar&RushOrder=adNumeric&ReorderReson=adVarChar&ReorderReasonOther=adVarChar
```

Inputs

DELIVERYLOCATIONID

*Data Type:*adNumeric

Description:Specifies the ID of the location where the container is to be delivered.

Options:

*Required?:*Y

Table Name:INV_CONTAINER_ORDER *Database Field Name:*

DELIVERY_LOCATION_ID_FK

CONTAINERID

*Data Type:*adNumeric

Description:Specifies the ID of the container to be reordered.

Options:

*Required?:*Y

Table Name:IINV_CONTAINERS

*Database Field Name:*CONTAINER_ID

CONTAINERNAME

*Data Type:*adVarChar

Description:Specifies the name of the container to be reordered.

Options:

*Required?:*Y

Table Name:INV_CONTAINERS

*Database Field Name:*CONTAINER_NAME

COMMENTS

*Data Type:*adVarChar

Description:Specifies the comments associated with the container.

Options:

*Required?:*Y

Table Name:INV_CONTAINERS

Database Field Name:

CONTAINER_COMMENTS

OWNERID

*Data Type:*adVarChar

Description:Specifies the owner of the container.

Options:

*Required?:*Y

Table Name:INV_CONTAINERS

*Database Field Name:*OWNER_ID_FK

CURRENTUSERID

*Data Type:*adVarChar

Description:Specifies the current user of the container.

Options:

*Required?:*Y

Table Name:INV_CONTAINERS

Database Field Name:

CURRENT_USER_ID_FK

NUMCOPIES

*Data Type:*adNumeric

Description:Specifies the number of container copies to be ordered.

Options:

*Required?:*Y

Table Name:N/A

*Database Field Name:*N/A

PROJECT

*Data Type:*adVarChar

Description:Specifies the project under which the container is to be ordered.

Options:

*Required?:*Y

Table Name:INV_CONTAINER_ORDER

*Database Field Name:*PROJECT_NO

JOB

*Data Type:*adVarChar

Description:Specifies the job associated with

Options:

*Required?:*the project under which the container is ordered.

Y

Table Name:INV_CONTAINER_ORDER

*Database Field Name:*JOB_NO

RUSHORDER

*Data Type:*adNumeric

Description:Specifies whether or not it is a rush order. 1 indicates that the order is rush order, 0 indicates that the order is not a rush order.

Options:1|0

*Required?:*Y

Table Name:INV_CONTAINER_ORDER

Database Field Name:

ISRUSHORDER

DUEDATE

*Data Type:*Date

Description:Specifies the date on which the order should be delivered. The date is specified in the date-time literal format. For example 01-27-2001.

Options:

*Required?:*Y

Table Name:INV_CONTAINER_ORDER

*Database Field Name:*DUE_DATE

REORDERREASON

*Data Type:*adVarChar

Description:Specifies the reason for reordering the container.

Options:

*Required?:*Y

Table Name:N/A

*Database Field Name:*N/A

REORDERREASONOTHER

*Data Type:*adVarChar

Description:Specifies the reason for reordering container, if the reason is other than those defined already in Inventory Enterprise.

Options:

*Required?:*Y, If Other is specified in ReorderReason.

Table Name:N/A

*Database Field Name:*N/A

Output

The ID of the newly created container, which is stored in the NewContainerID parameter. The data type of the NewContainerID parameter is adNumeric.

Example

The following code reorders a container with ContainerID 1764:

```
/cheminv/api/ReorderCon-
tainer.asp?DeliveryLocationID=1&Con-
tainerID=1764&ContainerName=nitroben
zene&Comments=adVarChar&Own-
erID=adVarChar&CurrentUserID=Sys-
```

```
tem&NumCopies=1&Project=2&Job=1&Rush
Order=1&ReorderReson=Insufficient
Quantity
```

RetireContainer

Retires a container and sets the status and location of the container to the specified values. When a container is retired its contents are not disposed, if specified explicitly.

Syntax

```
/cheminv/api/RetireContainer.asp?ContainerID=adNumeric&LocationID=adNumeric&ContainerStatusID=adNumeric&QtyRemaining=adDouble
```

Inputs

CONTAINERID

*Data Type:*adNumeric

Description:Specifies the ID of the container to be retired.

Options:

*Required?:*Y

Table Name:INV_CONTAINERS

*Database Field Name:*CONTAINER_ID

LOCATIONID

*Data Type:*adNumeric

Description:Specifies the ID of the location where the retired container needs to be moved. By default, the retired container is moved to the Disposed location.

Options:

*Required?:*Y

Table Name:INV_CONTAINERS

*Database Field Name:*LOCATION_ID_FK

CONTAINERSTATUSID

*Data Type:*adNumeric

Description:Specifies the ID of the status to be used for the retired container. By default, the retired container is set to the Disposed status.

Options:

*Required?:*Y

Table Name:INV_CONTAINERS *Database Field Name:*CONTAINER_STATUS_ID_FK

QTYREMAINING

*Data Type:*adDouble

Description:Specifies the quantity to be left in the container when it is retired. By default, the QtyRemaining parameter is set to zero.

Options:

*Required?:*N

Table Name:INV_CONTAINERS

*Database Field Name:*QTY_REMAINING

Output

None.

Example

The following code retires a container with ID 12345, and set both, its location and status to Disposed. The ID of the Disposed location is 2 and the ID of the Disposed status is 6:

```
/cheminv/api/RetireContainer.asp?ContainerID=12345&LocationID=2&ContainerStatusID=6
```

RetirePlate

Retires a plate and sets the status and location of the plate to the specified values. When a plate is retired its contents are not disposed, if specified explicitly.

Syntax

```
/cheminv/api/RetirePlate.asp?Plate_ID=adVarChar&Location_ID_FK=adDouble&Status_ID_FK=adNumeric&Qty_Remaining=adDouble
```

Inputs

PLATE_ID

*Data Type:*adVarChar

Description:Specifies the ID of the plate to be retired.

Options:

*Required?:*Y

Table Name:INV_PLATES

*Database Field Name:*PLATE_ID

QTY_REMAINING

*Data Type:*adDouble

Description:Specifies the quantity to be left in the plate after it is retired.

Options:

*Required?:*Y

Table Name:INV_PLATES

*Database Field Name:*QTY_REMAINING

STATUS_ID_FK

*Data Type:*adNumeric

Description:Specifies the ID of the status to be used for the retired plate. By default, the retired plate is set to the Disposed status.

Options:

*Required?:*Y

Table Name:INV_PLATES

*Database Field Name:*STATUS_ID_FK

LOCATION_ID_FK

*Data Type:*adDouble

Description:Specifies the ID of the location where the retired plate is to be moved. By default, the retired plate is moved to the Disposed location.

Options:

*Required?:*Y

Table Name: INV_PLATES

*Database Field Name:*LOCATION_ID_FK

Output

None.

Example

The following code retires a plate with ID P1008:

```
/cheminv/api/RetirePlate.asp?Plate_ID=P1008&Location_ID_FK=1212&Status_ID_FK=5&Qty_Remaining=0.001
```

ShipOrder

Ships order(s) to the specified locations and sets the status of the order(s) to Shipped.

Syntax

```
/cheminv/api/ShipOrder.asp?ShippedOrderIDList=adVarChar
```

Inputs

SHIPPEDORDERIDLIST

*Data Type:*adVarChar

Description:Specifies a comma delimited list of IDs of the orders to be shipped.

Options:

*Required?:*Y

Table Name:INV_ORDERS

*Database Field Name:*ORDER_ID

Output

The ID of the shipped order, which is stored in the OrderID parameter. The data type of the OrderID parameter is adNumeric.

Example

The following code ships the orders with IDs 1032 and 1007:

```
/cheminv/api/ShipOrder.asp?ShippedOrderIDList=1032,1007
```

SolvatePlate

Solvates a plate(s) with the specified type and amount of solvent.

Syntax

```
/cheminv/api/SolvatePlate.asp?PlateIDList=adVarChar&SolventIDList=adVarChar&SolventVolumeList=adVarChar&SolventVolumeUnitIDList=adVarChar&ConcentrationList=adVarChar&ConcentrationUnitIDList=adVarChar
```

Inputs

PLATEIDLIST

*Data Type:*adVarChar

Description:Specifies a comma delimited list of the IDs of the plates to be solvated.

Options:

*Required?:*Y

Table Name:INV_PLATES

*Database Field Name:*PLATE_ID

SOLVENTIDLIST

*Data Type:*adVarChar

Description:Specifies a comma delimited list of IDs of the solvents to be used for solvating plates.

Options:

*Required?:*Y

Table Name:INV_PLATES

*Database Field Name:*SOLVENT_ID_FK

SOLVENTVOLUMELIST

*Data Type:*adVarChar

Description:Specifies a comma delimited list of the volumes of the solvents.

Options:

*Required?:*Y

Table Name:INV_PLATES

*Database Field Name:*SOLVENT_VOLUME

SOLVENTVOLUMEUNITIDLIST

*Data Type:*adVarChar

Description:Specifies a comma delimited list of the IDs of the units in which the solvent volumes are to be measured.

Options:

*Required?:*Y

Table Name:IINV_PLATES

Database Field Name:

SOLVENT_VOLUME_UNIT_ID_FK

CONCENTRATIONLIST

*Data Type:*adVarChar

Description:Specifies a comma delimited list of the concentrations of the plates.

Options:

*Required?:*Y

Table Name:INV_PLATES

*Database Field Name:*CONCENTRATION

CONCENTRATIONUNITIDLIST

*Data Type:*adVarChar

Description:Specifies a comma delimited list of the IDs of the units in which the concentrations are to be measured.

Options:

*Required?:*Y

Table Name:INV_PLATES

*Database Field Name:*CONC_UNIT_FK

Output

None.

Example

The following code solvates the P1234 plate with a solvent with ID 120:

```
/cheminv/api/SolvatePlate.asp?PlateIDList=P1234&SolventIDList=120&SolventVolumeList=10&SolventVolumeUnitIDList=4&ConcentrationList=.1&ConcentrationUnitIDList=4
```

UndoDelivery

Undoes the delivery of a request and moves the container associated with the request back to its original location.

Syntax

```
/cheminv/api/UndoDelivery.asp?RequestID=adNumeric
```

Input

REQUESTID

*Data Type:*adNumeric

Description:Specifies the ID of the request, whose delivery is to be undone.

Options:

*Required?:*Y

Table Name:INV_REQUESTS

*Database Field Name:*REQUEST_ID

OUTPUT

None.

EXAMPLE

The following code undoes the delivery of a request with ID 12323:

```
/cheminv/api/UndoDelivery.asp?RequestID=12323
```

UpdateAddress

Updates the address of the specified supplier or location.

Syntax

```
/cheminv/api/UpdateAddress.asp?TableName=adVarChar&TablePKID=adVarChar[&AddressID=adVarChar&ContactName=adVarChar&Address1=adVarChar&Address2=adVarChar&Address3=adVarChar&Address4=adVarChar&City=adVarChar&StateIDFK=adNumeric&CountryIDFK=adNumeric&ZIP=adVarChar&FAX=adVarChar&Phone=adVarChar&Email=adVarChar]
```

Inputs

TABLENAME

*Data Type:*adVarChar

Description:Specifies the table the address associated with which is to be updated.

Options:

Options:

INV_SUPPLIERS| INV_LOCATIONS

*Required?:*Y

Table Name:N/A

*Database Field Name:*N/A

TABLEPKID

*Data Type:*adVarChar

Description:Specifies the primary key ID. This ID specifies the location or supplier, whose address is to be updated.

Options:

*Required?:*Y

Table Name:N/A

*Database Field Name:*N/A

ADDRESSID

*Data Type:*adVarChar

Description:Specifies the ID of the address to be updated.

Options:

*Required?:*N

Table Name:INV_ADDRESS

*Database Field Name:*ADDRESS_ID

CONTACTNAME

*Data Type:*adVarChar

Description:Specifies the contact name.

Options:

*Required?:*N

Table Name:INV_ADDRESS

*Database Field Name:*CONTACT_NAME

ADDRESS1

*Data Type:*adVarChar

Description:Specifies the first line of the address.

Options:

*Required?:*N

Table Name:INV_ADDRESS

*Database Field Name:*ADDRESS1

ADDRESS2

*Data Type:*adVarChar

Description:Specifies the second line of the address.

Options:

*Required?:*N

Table Name:INV_ADDRESS

*Database Field Name:*ADDRESS2

ADDRESS3

*Data Type:*adVarChar

Description:Specifies the third line of the address.

Options:

*Required?:*N

Table Name:INV_ADDRESS

*Database Field Name:*ADDRESS3

ADDRESS4

*Data Type:*adVarChar

Description:Specifies the fourth line of the address.

Options:

*Required?:*N
Table Name:INV_ADDRESS
*Database Field Name:*ADDRESS4

CITY

*Data Type:*adVarChar
Description:Specifies the name of the city.
Options:
*Required?:*N
Table Name:INV_ADDRESS
*Database Field Name:*CITY

STATEIDFK

*Data Type:*adNumeric
Description:Specifies the ID of the state.
Options:
*Required?:*N
Table Name:INV_ADDRESS
*Database Field Name:*STATE_ID_FK

COUNTRYIDFK

*Data Type:*adNumeric
Description:Specifies the ID of the country.
Options:
*Required?:*N
Table Name:INV_ADDRESS
*Database Field Name:*COUNTRY_ID_FK

ZIP

*Data Type:*adVarChar
Description:Specifies the zip code.
Options:
*Required?:*N
Table Name:INV_ADDRESS
*Database Field Name:*ZIP

FAX

*Data Type:*adVarChar
Description:Specifies the fax number.
Options:
*Required?:*N
Table Name:INV_ADDRESS
*Database Field Name:*FAX

PHONE

*Data Type:*adVarChar
Description:Specifies the phone number.
Options:
*Required?:*N
Table Name:INV_ADDRESS
*Database Field Name:*PHONE

EMAIL

*Data Type:*adVarChar
Description:Specifies the email address.
Options:
*Required?:*N
Table Name:INV_ADDRESS
*Database Field Name:*EMAIL

Output

The ID of the updated address, which is stored in the AddressID parameter.

Example

The following code updates the address of a supplier:

```
/cheminv/api/UpdateAddress.asp?TableName=INV_SUPPLIERS&TablePKID=12
```

UpdateAllContainerFields

Updates attributes of a container.

CAUTION

If you remove an optional parameter from the API call, the parameter is set to NULL.

Syntax

```
/cheminv/api/UpdateAllContainerFields.asp?ContainerID=adNumeric&LocationID=adNumeric&UOMID=adNumeric&QtyMax=adDouble&ContainerTypeID=adNumeric[&ContainerStatusID=adNumeric&QtyRemaining=adDouble&CompoundID=adNumeric&RegID=adNumeric&BatchNumber=adNumeric&ExpDate=adDBTimeStamp&DateCertified=adDBTimeStamp&DateApproved=adDBTimeStamp&MinStockQty=adDouble&MaxStockQty=adDouble&ContainerName=adVarChar&ContainerDesc=adVarChar&TareWeight=adDouble&NetWeight=adDouble&FinalWeight=adDouble&UOWID=adNumeric&Purity=adDouble&UOPID=adNumeric&Concentration=adDouble&Density=adDouble&UOCID=adNumeric&UODID=adNumeric&Grade=adVarChar&SolventIDFK=adVarChar&StorageConditions=adVarChar&HandlingProcedures=adVarChar&Comments=adVarChar&SupplierID=adNumeric&SupplierCatNum=adVarChar&LotNum=adVarChar&DateProduced=adDBTimeStamp&DateOrdered=adDBTimeStamp&DateReceived=adDBTimeStamp&ContainerCost=adNumeric&UOCostID=adNumeric&&OwnerID=adVarChar&CurrentUserID=adVarChar&PONumber=adVarChar&POLineNumber=adVarChar&ReqNumber=adVarChar]
```

Inputs

CONTAINERID

*Data Type:*adNumeric

Description:Specifies the ID of the container that is to be updated.

Options:

*Required?:*Y

Table Name:INV_CONTAINERS

*Database Field Name:*CONTAINER_ID

LOCATIONID

*Data Type:*adNumeric

Description:Specifies the ID of the location where the container is to be stored.

Options:

*Required?:*Y

Table Name:IINV_CONTAINERS

*Database Field Name:*LOCATION_ID_FK

UOMID

*Data Type:*adNumeric

Description:Specifies the ID of the unit in which the contents of the container are to be measured. You can determine the ID of an unit using ShowContainerUnits.

Options:

*Required?:*Y

Table Name:INV_CONTAINERS

Database Field Name:

UNIT_OF_MEAS_ID_FK

QTYMAX

*Data Type:*adDouble

Description:Specifies the maximum quantity of the substance that the container can hold.

Options:

*Required?:*Y

Table Name:INV_CONTAINERS

*Database Field Name:*QTY_MAX

CONTAINERTYPEID

*Data Type:*adNumeric

Description:Specifies the ID of the type of the container. You can determine the values that ContainerTypeID parameter can take using ShowContainerTypes.

Options:

*Required?:*Y

Table Name:INV_CONTAINERS

Database Field Name:

CONTAINER_TYPE_ID_FK

CONTAINERSTATUSID

*Data Type:*adNumeric

Description:Specifies the ID of the status of the container. You can determine the values that ContainerStatusID parameter can take using ShowContainerStatus.

Options:

*Required?:*N

Table Name:INV_CONTAINERS

Database Field Name:

CONTAINER_STATUS_ID_FK

QTYREMAINING

*Data Type:*adDouble

Description:Specifies the amount of substance left in the container.

Options:

*Required?:*N

Table Name:INV_CONTAINERS

*Database Field Name:*QTY_REMAINING

BARCODE

*Data Type:*adVarChar

Description:Specifies the barcode of the container. The container barcode is an alphanumerical identifier used to identify the container uniquely.

Options:

*Required?:*N

Table Name:INV_CONTAINERS

*Database Field Name:*BARCODE

COMPOUNDID

*Data Type:*adNumeric

Description:Specifies the ID of the substance that is to be stored in the container.

Options:

*Required?:*N

Table Name:INV_CONTAINERS

*Database Field Name:*COMPOUND_ID_FK

REGID

*Data Type:*adNumeric

Description:Specifies the registration ID of the substance that is to be stored in the container.

Options:

*Required?:*N

Table Name:INV_CONTAINERS

*Database Field Name:*REG_ID_FK

BATCHNUMBER

Data Type: adNumeric

Description:Specifies the batch number associated with the registration ID of the substance stored in the container.

Options:

*Required?:*N

Table Name:INV_CONTAINERS

*Database Field Name:*BATCH_NUMBER_FK

EXPDATE

*Data Type:*adDBTimeStamp

Description:Specifies the expiration date of the contents of the container.

Options:

*Required?:*N

Table Name:INV_CONTAINERS

*Database Field Name:*DATE_EXPIRES

DATECERTIFIED

*Data Type:*adDBTimeStamp

Description:Specifies the date the container was certified.

Options:

*Required?:*N

Table Name:INV_CONTAINERS

*Database Field Name:*DATE_CERTIFIED

DATEAPPROVED

*Data Type:*adDBTimeStamp

Description:Specifies the date the certification of the container was approved.

Options:

*Required?:*N

Table Name:INV_CONTAINERS

*Database Field Name:*DATE_APPROVED

MINSTOCKQTY

*Data Type:*adDouble

Description:Specifies the minimum stock quantity for the contents of the container. Minimum stock quantity is the minimum quantity of the substance that stock should always hold. The unit for the minimum stock quantity is same as the value of the UOMID parameter. Inventory Enterprise issues a re-stock warning if the quantity of the substance in the stock is less than the value specified for the MinStockQty parameter.

Options:

*Required?:*N

Table Name:INV_CONTAINERS

*Database Field Name:*QTY_MINSTOCK

MAXSTOCKQTY

*Data Type:*adDouble

Description:Specifies the maximum stock quantity for the contents of the container. Maximum stock quantity is the maximum quantity of the substance that stock can hold. The unit for the maximum stock quantity is same as the value of the UOMID parameter. Inventory Enterprise issues a over-stock warning if the quantity of the substance in the stock exceeds than the value of the MaxStockQty parameter.

Options:

*Required?:*N

Table Name:INV_CONTAINERS

*Database Field Name:*QTY_MAXSTOCK

CONTAINERNAME

*Data Type:*adVarChar

Description:Specifies the name of the container. If a value for the ContainerName parameter is not specified, by default, it is set to the internal container ID.

Options:

*Required?:*N

Table Name:INV_CONTAINERS

*Database Field Name:*CONTAINER_NAME

CONTAINERDESC

*Data Type:*adVarChar

Description:Specifies the description of the container.

Options:

*Required?:*N

Table Name:INV_CONTAINERS

Database Field Name:
CONTAINER_DESCRIPTION

TAREWEIGHT

*Data Type:*adDouble

Description:Specifies the tare weight of the container. The unit for the tare weight of the container is specified in the UOWID parameter.

Options:

*Required?:*N

Table Name:INV_CONTAINERS

*Database Field Name:*TARE_WEIGHT

NETWEIGHT

*Data Type:*adDouble

Description:Specifies the net weight of the container. The unit for the net weight of the container is specified in the UOWID parameter.

Options:

*Required?:*N

Table Name:INV_CONTAINERS

*Database Field Name:*NET_WGHT

FINALWEIGHT

*Data Type:*adDouble

Description:Specifies the final weight of the container. The unit for the final weight of a container is specified in the UOWID parameter.

Options:

Required?: N

Table Name:INV_CONTAINERS

*Database Field Name:*FINAL_WGHT

UOWID

*Data Type:*adNumeric

Description:Specifies the ID of the unit in which the tare, net, and final weights of the container are measured. You can determine the values that the UOWID parameter can take using ShowContainerUnits.

Options:

*Required?:*N

Table Name:INV_CONTAINERS

Database Field Name:
UNIT_OF_WGHT_ID_FK

PURITY

*Data Type:*adDouble

Description: Specifies the purity of the substance specified in the CompoundID parameter. The unit for the value of the Purity parameter is specified in the UOPID parameter.

Options:

*Required?:*N

Table Name:INV_CONTAINERS

*Database Field Name:*PURITY

UOPID

*Data Type:*adNumeric

Description:Specifies the ID of the unit in which the purity of the container contents is measured. You can determine the values that the UOPID parameter can take using ShowContainerUnits.

Options:

*Required?:*N

Table Name:INV_CONTAINERS

Database Field Name:

UNIT_OF_PURITY_ID_FK

CONCENTRATION

*Data Type:*adDouble

Description:Specifies the concentration of the substance stored in the container. The unit of the concentration is specified in the UOCID parameter.

Options:

*Required?:*N

Table Name:INV_CONTAINERS

*Database Field Name:*CONCENTRATION

DENSITY

*Data Type:*adDouble

Description:Specifies the density of the substance stored in the container.

Options:

*Required?:*N

Table Name:INV_CONTAINERS

*Database Field Name:*DENSITY

UOCID

*Data Type:*adNumeric

Description:Specifies the ID of the unit in which the concentration of the container contents is measured. You can determine the values that the UOCID parameter can take using ShowContainerUnits.

Options:

*Required?:*N

Table Name:INV_CONTAINERS

*Database Field Name:*UNIT_OF_COST_ID_FK

UODID

*Data Type:*adNumeric

Description:Specifies the ID of the unit in which the density of the container contents is measured. You can determine the values that the UODID parameter can take using ShowContainerUnits.

Options:

*Required?:*N

Table Name:INV_CONTAINERS

Database Field Name:
UNIT_OF_DENSITY_ID_FK

GRADE

*Data Type:*adVarChar

Description:Specifies the grade associated with quality or purity of the container contents.

Options:

*Required?:*N

Table Name:INV_CONTAINERS

*Database Field Name:*GRADE

SOLVENTIDFK

*Data Type:*adVarChar

Description:Specifies the solvent for the contents of the new container.

Options:

*Required?:*N

Table Name:INV_CONTAINERS

*Database Field Name:*SOLVENT_ID_FK

COMMENTS

*Data Type:*adVarChar

Description:Specifies the comments or additional information associated with the container. The Comments parameter can take comments up to 4000 characters. The Comments parameter can be passed as form value, so that it is not truncated if the size of the query string size exceeds the specified limit.

Options:

Required?: N

Table Name:INV_CONTAINERS

Database Field Name:

CONTAINER_COMMENTS

STORAGECONDITIONS

*Data Type:*adVarChar

Description:Specifies the storage conditions for the container.

Options:

*Required?:*N

Table Name:INV_CONTAINERS

Database Field Name:

STORAGE_CONDITIONS

HANDLINGPROCEDURES

*Data Type:*adVarChar

Description:Specifies handling procedures for the container.

Options:

Required?: N

Table Name:INV_CONTAINERS

Database Field Name:

HANDLING_PROCEDURES

SUPPLIERID

*Data Type:*adNumeric

Description:Specifies the ID of the supplier of the container. You can determine the values that the SupplierID parameter can take using ShowSuppliers.

Options:

*Required?:*N

Table Name:INV_CONTAINERS

*Database Field Name:*SUPPLIER_ID_FK

SUPPLIERCATNUM

*Data Type:*adVarChar

Description:Specifies the catalog number that supplier of the container has associated with the container.

Options:

*Required?:*N

Table Name:INV_CONTAINERS

*Database Field Name:*SUPPLIER_CATNUM

LOTNUM

*Data Type:*adVarChar

Description:Specifies the lot number of the supplier of the container.

Options:

*Required?:*N

Table Name:INV_CONTAINERS

*Database Field Name:*LOT_NUM

DATEPRODUCED

*Data Type:*adDBTimeStamp

Description:Specifies the date the contents of the container were produced. For example, 01-27-2006.

Options:

*Required?:*N

Table Name:INV_CONTAINERS

*Database Field Name:*DATE_PRODUCED

DATEORDERED

*Data Type:*adDBTimeStamp

Description:Specifies the date the container was ordered. For example, 01-27-2006.

Options:

*Required?:*N

Table Name:INV_CONTAINERS

*Database Field Name:*DATE_ORDERED

DATERECIEVED

*Data Type:*adDBTimeStamp

Description:Specifies the date the container was received. For example, 01-27-2006.

Options:

*Required?:*N

Table Name:INV_CONTAINERS

Database Field Name: DATE_RECEIVED

CONTAINERCOST

*Data Type:*adNumeric

Description:Specifies the cost of the new container. The unit of the cost of the container is USD, by default.

Options:

*Required?:*N

Table Name:INV_CONTAINERS

*Database Field Name:*CONTAINER_COST

UOCOSTID

*Data Type:*adNumeric

Description:Specifies the unit of the cost of the container. By default, the unit of the cost of the container is USD.

Options:

*Required?:*N

Table Name:INV_CONTAINERS

*Database Field Name:*UNIT_OF_COST_ID_FK

OWNERID

*Data Type:*adVarChar

Description:Specifies the ID of the owner of the container.

Options:

*Required?:*N

Table Name:INV_CONTAINERS

*Database Field Name:*OWNER_ID_FK

CURRENTUSERID

*Data Type:*adVarChar

Description:Specifies the ID of the user of the container.

Options:

*Required?:*N

Table Name:INV_CONTAINERS

Database Field Name:
CURRENT_USER_ID_FK

PONUMBER

*Data Type:*adVarChar

Description:Specifies the PO number under which the container was ordered.

Options:

*Required?:*N

Table Name:INV_CONTAINERS

*Database Field Name:*PO_NUMBER

POLINENUMBER

*Data Type:*adVarChar

Description:Specifies the PO line number of the container.

Options:

*Required?:*N

Table Name:INV_CONTAINERS

*Database Field Name:*PO_LINE_NUMBER

REQNUMBER

*Data Type:*adVarChar

Description:Specifies the internal requisition number under which the container was ordered.

Options:

*Required?:*N

Table Name:INV_CONTAINERS

*Database Field Name:*REQ_NUMBER

FIELD_1

*Data Type:*adVarChar

Description:Specifies the value of the configurable field Field_1.

Options:

*Required?:*N

Table Name:INV_CONTAINERS

*Database Field Name:*FIELD_1

FIELD_2

*Data Type:*adVarChar

Description:Specifies the value of the configurable field Field_2.

Options:

*Required?:*N

Table Name:INV_CONTAINERS

*Database Field Name:*FIELD_2

FIELD_3

*Data Type:*adVarChar

Description:Specifies the value of the configurable field Field_3.

Options:

*Required?:*N

Table Name:INV_CONTAINERS

*Database Field Name:*FIELD_3

FIELD_4

*Data Type:*adVarChar

Description:Specifies the value of the configurable field Field_4.

Options:

*Required?:*N

Table Name:INV_CONTAINERS

*Database Field Name:*FIELD_4

FIELD_5

*Data Type:*adVarChar

Description:Specifies the valueof the configurable field Field_5.

Options:

*Required?:*N

Table Name:INV_CONTAINERS

*Database Field Name:*FIELD_5

FIELD_6

*Data Type:*adVarChar

Description:Specifies the value of the configurable field Field_6.

Options:

*Required?:*N

Table Name:INV_CONTAINERS

*Database Field Name:*FIELD_6

FIELD_7

*Data Type:*adVarChar

Description:Specifies the value of the configurable field Field_7.

Options:

*Required?:*N

Table Name:INV_CONTAINERS

*Database Field Name:*FIELD_7

FIELD_8

*Data Type:*adVarChar

Description:Specifies the value of the configurable field Field_8.

Options:

*Required?:*N

Table Name:INV_CONTAINERS

*Database Field Name:*FIELD_8

FIELD_9

*Data Type:*adVarChar

Description:Specifies the value of the configurable field Field_9.

Options:

*Required?:*N

INV_CONTAINERS

*Database Field Name:*FIELD_9

FIELD_10

*Data Type:*adVarChar

Description:Specifies the value of the configurable field Field_10.

Options:

*Required?:*N

Table Name:INV_CONTAINERS

*Database Field Name:*FIELD_10

DATE_1

*Data Type:*adDBTimeStamp

Description:Specifies the value of the configurable field Date_1.

Options:

*Required?:*N

Table Name:INV_CONTAINERS

*Database Field Name:*DATE_1

DATE_2

*Data Type:*adDBTimeStamp

Description:Specifies the value of the configurable field Date_2.

Options:

*Required?:*N

Table Name:INV_CONTAINERS

*Database Field Name:*DATE_2

DATE_3

*Data Type:*adDBTimeStamp

Description:Specifies the value of the configurable field Date_3.

Options:

*Required?:*N

Table Name:INV_CONTAINERS

*Database Field Name:*DATE_3

DATE_4

*Data Type:*adDBTimeStamp

Description:Specifies the value of the configurable field Date_4.

Options:

*Required?:*N

Table Name:INV_CONTAINERS

*Database Field Name:*DATE_4

Date_5

*Data Type:*adDBTimeStamp

Description:Specifies the value of the configurable field Date_5.

Options:

*Required?:*N

INV_CONTAINERS

*Database Field Name:*DATE_5

Output

The ID of the updated container, which is stored in the ContainerID parameter.

Example

The following code updates the attributes of a container with ID C1001:

```
/cheminv/api/UpdateAllContainer-
Fields.asp?ContainerID=C1001&Loca-
tionID=101&UOMID=6&QtyMax=500&Contai
nerTypeID=3
```

UpdateContainer2

Updates the specified attributes of the container(s).

Syntax

```
/cheminv/api/
UpdateContainer2.asp?Contain-
erIDs=adVarChar&ValuePairs=adVarChar
```

Inputs

CONTAINERIDS

*Data Type:*adVarChar

Description:Specifies a ì::î list of the IDs of the containers to be updated.

Options:

*Required?:*Y

Table Name:INV_CONTAINERS

*Database Field Name:*CONTAINER_ID

VALUEPAIRS

*Data Type:*adVarChar

Description:Specifies a comma delimited list of the pairs of the name and value of the parameters to be updated.

For example: colnam1%3Dval1,colnam2%3Dval2

Options:

*Required?:*Y

Table Name:N/A

*Database Field Name:*N/A

Output

None.

Example

The following code updates the qty_remaining and container_description fields of the containers with IDs 212 and 213:

```
Updatecontainer2.asp?Contain-
erIDs=212,213&Value-
Pairs=qty_remaining%3D5,container_de
scription%3Dupdated
```

UpdateContainerQtyRemaining

<FONTSIZE=3Updates the remaining quantity for a container.

Syntax

```
/cheminv/api/UpdateContainerQtyRe-
maining.asp?ContainerID=adNumer-
ic&QtyRemaining=adDouble
```

Inputs

CONTAINERID

*Data Type:*adNumeric

Description:Specifies the ID of the container, whose remaining quantity is to be updated.

Options:

*Required?:*Y

Table Name:INV_CONTAINERS

*Database Field Name:*CONTAINER_ID

QTYREMAINING

*Data Type:*adDouble

Description:Specifies the quantity of the substance left in the container. The unit of the remaining quantity is same as the unit defined in the UOMID parameter.

Options:

*Required?:*Y

Table Name:INV_CONTAINERS

*Database Field Name:*QTY_REMAINING

Output

The ID of the updated container, which is stored in the ContainerID parameter.

Example

The following code updates remaining quantity for a container with ID C1001:

```
/cheminv/api/UpdateContainerQtyRemaining.asp?ContainerID=C1001&QtyRemaining=10
```

UpdateGridFormat

Updates a grid or storage format.

Syntax

```
/cheminv/api/UpdateGridFormat.asp?grid_format_id=adNumeric&name=adVarChar&row_count=adNumeric&col_count=adNumeric&zero_padding_count=adNumeric[&row_prefix=adVarChar&col_prefix=adVarChar&row_use_letters=adNumeric&col_use_letters=adNumeric&name_separator=adVarChar&number_start_corner=adNumeric&number_direction=adNumeric&description=adVarChar&grid_format_type=adNumeric]
```

Inputs

GRID_FORMAT_ID

*Data Type:*adNumeric

Description:Specifies the ID of the grid to be updated.

Options:

*Required?:*Y

Table Name:INV_GRID_FORMAT

*Database Field Name:*GRID_FORMAT_ID

GRID_FORMAT_TYPE

*Data Type:*adNumeric

Description:Specifies the type of the grid format.

Options:

*Required?:*N

Table Name:INV_GRID_FORMAT

Database Field Name:

GRID_FORMAT_TYPE_FK

ROW_COUNT

*Data Type:*adNumeric

Description:Specifies the number of rows that the grid should contain.

Options:

*Required?:*Y

Table Name:INV_GRID_FORMAT

Database Field Name:

ROW_COUNT

COL_COUNT

*Data Type:*adNumeric

Description:Specifies the number of columns that the grid should contain.

Options:

*Required?:*Y

Table Name:INV_GRID_FORMAT

Database Field Name:

COL_COUNT

ROW_PREFIX

*Data Type:*adVarChar

Description:Specifies the prefix to be used for the grid rows. N

Table Name:INV_GRID_FORMAT

Database Field Name:

ROW_PREFIX

COL_PREFIX

*Data Type:*adVarChar

Description:Specifies the prefix to be used for the grid columns.

Options:

*Required?:*N

Table Name:INV_GRID_FORMAT

Database Field Name:

COL_PREFIX

ROW_USE_LETTERS

*Data Type:*adNumeric

Description:Specifies whether to label rows with letters or with numbers.

Options:

*Required?:*N

Table Name:INV_GRID_FORMAT

Database Field Name:

ROW_USE_LETTERS

COL_USE_LETTERS

Data Type: adNumeric

Description:Specifies whether to label columns with letters or with numbers.

Options:

*Required?:*N

Table Name:INV_GRID_FORMAT

Database Field Name:

COL_USE_LETTERS

NAME_SEPARATOR

*Data Type:*adVarChar

Description:Specifies the character that separates the row and column IDs. For example, '-' is the name separator for the following column and row IDs: A-1 and B-1.

Options:

*Required?:*N

Table Name:INV_GRID_FORMAT

Database Field Name:

NAME_SEPARATOR

NUMBER_START_CORNER

*Data Type:*adNumeric

Description:Specifies the value with which the plate numbering is to be started.

Options:

*Required?:*N

Table Name:INV_GRID_FORMAT

Database Field Name:

NUMBER_START_CORNER

NUMBER_DIRECTION

*Data Type:*adNumeric

Description:Specifies whether to fill the rows first or to fill the columns first.

Options:

*Required?:*N

Table Name:INV_GRID_FORMAT

*Database Field Name:*NUMBER_DIRECTION

NAME

*Data Type:*adVarChar

Description:Specifies the name of the grid format.

Options:

*Required?:*Y

Table Name:INV_GRID_FORMAT

*Database Field Name:*NAME

DESCRIPTION

*Data Type:*adVarChar

Description:Specifies the description of the grid format.

Options:

*Required?:*N

Table Name:INV_GRID_FORMAT

*Database Field Name:*DESCRIPTION

ZERO_PADDING_COUNT

*Data Type:*adNumeric

Description:Specifies the number of digits with which the column and row numbers are to be padded. Usually, this parameter is set to 2.

Options:

*Required?:*Y

Table Name:INV_GRID_FORMAT

Database Field Name:

ZERO_PADDING_COUNT

Output

The ID of the updated grid format, which is stored in the new_grid_format_id parameter. The data type of the new_grid_format_id parameter is adNumeric.

Example

The following code updates a grid format with ID 14:

```
/cheminv/api/UpdateGridFor-
mat.asp?grid_format_id=14&name=TestG
rid&row_count=12&col_count=8&zero_pa
dding_count=1[&col_prefix=C]
```

UpdateLink

Updates the attributes of a link, such as URL, link text, or image source.

Syntax

```
/cheminv/api/Up-
dateLink.asp?FK_Value=adVar-
Char&FK_Name=adVarChar&Table_name=ad
VarChar[&URLID=Int&URLHref=adVar-
Char&ImageSource=adVarChar&Link-
Text=adVarChar]
```

Inputs

URLID

*Data Type:*adInteger

Description:Specifies the ID of the link to be updated.

Options:

*Required?:*N

Table Name:INV_URL

*Database Field Name:*URL_ID

FK_NAME

*Data Type:*adVarChar

Specifies the field that uniquely identifies the row with which the link is to be associated.

Options:

*Required?:*Y

Table Name:INV_URL

*Database Field Name:*FK_NAME

FK_VALUE

*Data Type:*adVarChar

Description:Specifies the value of the field specified in the FK_Name parameter.

Options:

*Required?:*Y

Table Name:INV_URL

*Database Field Name:*FK_VALUE

TABLE_NAME

*Data Type:*adVarChar

Description:Specifies the table name that contains the row with which the link is to be associated.

Options:

*Required?:*Y

Table Name:INV_URL

*Database Field Name:*TABLE_NAME

URLHREF

Data Type: adVarChar

Description:Specifies the URL of the link.

Options:

*Required?:*N

Table Name:INV_URL

*Database Field Name:*URL

URLTYPE

Data Type: adVarChar

Description:Specifies the type of the URL.

Options:

Required?: N

Table Name:INV_URL

*Database Field Name:*URL_TYPE

LINKTEXT

*Data Type:*adVarChar

Description:Specifies the text for the link. If no text is specified, the link text is set to the URL of the link.

Options:

Required?: N

Table Name: INV_URL

*Database Field Name:*LINK_TXT

IMAGESOURCE

*Data Type:*adVarChar

Description:Specifies the path of the GIF or JPEG image to be used for the link.

Options:

*Required?:*N

Table Name:INV_URL

*Database Field Name:*IMAGE_SRC

Output

The ID of the updated link. This ID is stored in the URLID parameter, the data type of which is adNumeric.

Example

The following code updates the LinkText parameter of an already created link:

```
/cheminv/api/
UpdateLink.asp?FK_Value=1500&FK_Name
```

```
=CONTAINER_ID&Table_name=INV_CONTAIN
ERS[&LinkText="Search Engine"]
```

UpdatePhysPlateType

Updates a physical plate type.

Syntax

```
/cheminv/api/UpdatePhysPlate-
Type.asp?PhysPlateID=adNumeric&Grid-
FormatID=adNumeric&PhysPlateName=adV
arChar&RowCount=adNumeric&Col-
Count=adNumeric&WellCapacity=Int&Ca-
pacityUnitID=adNumeric&ZeroPaddingCo
unt=adNumeric[&RowPrefix=adVar-
Char&ColPrefix=adVarChar&RowUseLet-
ters=adNumeric&ColUseLetters=adNumer
ic&NameSeparator=adVarChar&Number-
StartCorner=adNumeric&NumberDirec-
tion=adNumeric&IsPreBarcoded=adVarCh
ar&SupplierID=adNumeric]
```

Inputs

PHYSPLATEID

*Data Type:*adNumeric

Description:Specifies the ID of the physical plate type to be updated.

Options:

*Required?:*Y

Table Name:INV_PHYSICAL_PLATE

*Database Field Name:*PHYS_PLATE_ID

GRIDFORMATID

*Data Type:*adNumeric

Description:Specifies the ID of the grid format associated with the physical plate type.

Options:

*Required?:*Y

Table Name:INV_PHYSICAL_PLATE *Database Field Name:*GRID_FORMAT_ID_FK

PHYSPLATENAME

*Data Type:*adVarChar

Description:Specifies the name of the physical plate type.

Options:

*Required?:*Y

Table Name:INV_PHYSICAL_PLATE *Database Field Name:*PHYS_PLATE_NAME

ROWCOUNT

*Data Type:*adNumeric

Description:Specifies the number of rows that the plate created using the new plate type will contain.

Options:

*Required?:*Y

Table Name:INV_GRID_FORMAT

*Database Field Name:*ROW_COUNT

COLCOUNT

*Data Type:*adNumeric

Description:Specifies the number of columns that the plate created using the new plate type will contain.

Options:

*Required?:*Y

Table Name:INV_GRID_FORMAT

*Database Field Name:*COL_COUNT

ROWPREFIX

*Data Type:*adVarChar

Description:Specifies prefix for the plate rows.

Options:

*Required?:*N

Table Name:INV_GRID_FORMAT

*Database Field Name:*ROW_PREFIX

COLPREFIX

*Data Type:*adVarChar

Description:Specifies prefix for the plate columns.

Options:

*Required?:*N

Table Name:INV_GRID_FORMAT

*Database Field Name:*COL_PREFIX

ROWUSELETTERS

*Data Type:*adNumeric

Specifies whether the rows are to be identified with numbers or letters. The rows are identified with letters if pRowUseLetters is set to 1.

Options: 1|0

*Required?:*N

Table Name:INV_GRID_FORMAT

*Database Field Name:*ROW_USE_LETTERS

COLUSELETTERS

*Data Type:*adNumeric

Description:Specifies whether the columns are to be identified with numbers or letters. The columns are identified with letters if pColUseLetters is set to 1.

Options: 1|0

*Required?:*N

Table Name:INV_GRID_FORMAT

*Database Field Name:*COL_USE_LETTERS

NAMESEPARATOR

Data Type: adVarChar

Description:Specifies the character that separates the row and column IDs. For example, '-' is the name separator for the following column and row IDs: A-1 and B-1.

Options:

*Required?:*N

Table Name:INV_GRID_FORMAT

*Database Field Name:*NAME_SEPARATOR

NUMBERSTARTCORNER

*Data Type:*adNumeric

Description:Specifies the value with which the plate numbering is to be started.

Options:

*Required?:*N

Table Name:INV_GRID_FORMAT

Database Field Name:
NUMBER_START_CORNER

NUMBERDIRECTION

Data Type: adNumeric

Description:Specifies whether to fill the rows first or to fill the columns first.

Options:

*Required?:*N

Table Name:INV_GRID_FORMAT

*Database Field Name:*NUMBER_DIRECTION

ISPREBARCODED

*Data Type:*adVarChar

Description:Specifies whether the barcodes should be already defined for the plates created using the new physical plate type. The barcodes are already defined for the plates if pIsPreBarcoded is set to Y.

Options: Y|N

*Required?:*N

Table Name:INV_PHYSICAL_PLATE

*Database Field Name:*IS_PRE_BARCODED

SUPPLIERID

*Data Type:*adNumeric

Description:Specifies ID of the suppliers of the plates created using the new physical plate type.

Options:

*Required?:*N

Table Name:INV_PHYSICAL_PLATE

*Database Field Name:*SUPPLIER_ID_FK

CAPACITYUNITID

*Data Type:*adNumeric

Description:Specifies the ID associated with the unit in which the well contents are to be measured.

Options:

*Required?:*Y

Table Name:INV_PHYSICAL_PLATE

Database Field Name:
CAPACITY_UNIT_ID_FK

WELLCAPACITY

*Data Type:*adNumeric

Description:Specifies the capacity of the wells of the plates created using the new physical plate type.

Options:

Required?: Y

Table Name:INV_PHYSICAL_PLATE

*Database Field Name:*WELL_CAPACITY

ZEROPADDINGCOUNT

*Data Type:*adNumeric

Description:Specifies the number of digits with which the column and row numbers are to be padded. Usually, this parameter is set to 2.

Options:

*Required?:*Y

Table Name:INV_GRID_FORMAT

Database Field Name:
ZERO_PADDING_COUNT

Output

The ID of the updated physical plate type, which is stored in the pPhysPlateID parameter. The data type of the pPhysPlateID parameter is adNumeric.

Example

The following code updates a physical plate type with ID 1020:

```
/cheminv/api/UpdatePhysPlate-
Type.asp?PhysPlateID=1020&GridForma-
tID=1021&PhysPlateName=TestPlate&Row
Count=8&ColCount=12&WellCapac-
ity=12&CapacityUnitID=4&ZeroPadding-
Count=2[&RowPrefix=B]
```

UpdatePlate

Updates the specified field(s) of the plate(s).

Syntax

```
/cheminv/api/Update-
Plate.asp?PlateIDs=adVarChar&Value-
Pairs=adVarChar
```

Inputs

PLATEIDS

*Data Type:*adVarChar

Description:Specifies a "::" delimited list of the IDs of the plates to be updated.

Options:

*Required?:*Y

Table Name:INV_PLATES

Database Field Name:

PLATE_ID

VALUEPAIRS

Data Type: adVarChar

Description:Specifies a "::" delimited list of the name and value pairs for the plate fields to be updated. For example: colnam1%3Dval1::colnam2%3Dval2, where "%3D" refers to "=".

Options:

*Required?:*Y

Table Name:N/A

*Database Field Name:*N/A

Output

None.

Example

The following code updates the Qty_Remaining and Solvent_Volume fields for plates with IDs P1148 and P1147:

```
/cheminv/api/UpdatePlate.asp?PlateIDs=P1148::P1147&ValuePairs=Qty_Remaining%3D10::Solvent_Volume%3D.2
```

UpdatePlateFormat

Updates a plate format.

Syntax

```
/cheminv/api/UpdatePlateFormat.asp?PlateFormatID=adNumeric&PlateFormatName=adVarChar&PhysPlateID=adNumeric
```

Inputs

PLATEFORMATID

*Data Type:*adNumeric

Description:Specifies the ID of the plate format to be updated.

Options:

*Required?:*Y

Table Name:INV_PLATE_FORMAT

*Database Field Name:*PLATE_FORMAT_ID

PLATEFORMATNAME

*Data Type:*adVarChar

Description:Specifies the name of the plate format.

Options:

*Required?:*Y

Table Name:INV_PLATE_FORMAT

Database Field Name:

PLATE_FORMAT_NAME

PHYSPLATEID

*Data Type:*adNumeric

Description:Specifies the ID of the physical plate type to be associated with the plate format.

Options:

*Required?:*Y

Table Name:INV_PLATE_FORMAT

*Database Field Name:*PHYS_PLATE_ID_FK

Output

The ID of the updated plate format. This ID is stored in the NewPlateFormatId parameter, the data type of which is adNumeric.

Example

The following code updates a plate format with ID 21:

```
/cheminv/api/UpdatePlateFormat.asp?PlateFormatID=21&PlateFormatName=NewPlateFormat&PhysPlateID=5
```

UpdatePlateType

Updates a plate type.

Syntax

```
/cheminv/api/UpdatePlateType.asp?PlateTypeID=adNumeric&PlateTypeName=adVarChar&MaxFreezeThaw=adNumeric
```

Input

PLATETYPEID

*Data Type:*adNumeric

Description:Specifies the ID of the plate type to be updated.

Options:

*Required?:*Y

Table Name:INV_PLATE_TYPES

*Database Field Name:*PLATE_TYPE_ID

PLATETYPENAME

*Data Type:*adVarChar

Description:Specifies the name of the plate type.

Options:

*Required?:*Y

Table Name:INV_PLATE_TYPES

*Database Field Name:*PLATE_TYPE_NAME

MAXFREEZETHAW

*Data Type:*adNumeric

Description:Specifies the maximum number of freeze thaw cycles allowed for the plate type.

Options:

*Required?:*Y

Table Name:INV_PLATE_TYPES

*Database Field Name:*MAX_FREEZE_THAW

Output

The ID of the updated plate type, which is stored in the PlatetypeID parameter.

Example

The following code updates a plate type with ID 4:

```
/cheminv/api/UpdatePlateType.asp?PlateTypeID=4&PlateTypeName=NewPlateType&MaxFreezeThaw=5
```

UpdateRequest

Updates a request.

Syntax

```
/cheminv/api/UpdateRequest.asp?RequestID=adNumeric&UserID=adVarChar&QtyRequired=adInteger&LocationID=adNumeric&ContainerTypeID=adNumeric&NumContainers=adNumeric&QtyList=adVarChar[&DateRequired=Date&RequestComments=adVarChar&RequestTypeID=adVarChar&ShipToName=adVarChar&ExpenseCenter=adVarChar]
```

Inputs

REQUESTID

*Data Type:*adNumeric

Description:Specifies the ID of the request to be updated.

Options:

*Required?:*Y

Table Name:INV_REQUESTS

*Database Field Name:*REQUEST_ID

USERID

*Data Type:*adVarChar

Description:Specifies the ID of the user who has requested the container.

Options:

*Required?:*Y

Table Name:INV_REQUESTS

*Database Field Name:*USER_ID_FK

QTYREQUIRED

*Data Type:*adInteger

Description:Specifies the quantity of the substance the requested container should contain.

Options:

*Required?:*Y

Table Name:INV_REQUESTS

*Database Field Name:*QTY_REQUIRED

LOCATIONID

*Data Type:*adNumeric

Description:Specifies the ID for the location where the request is to be delivered.

Options:

*Required?:*Y

Table Name:INV_REQUESTS

Database Field Name:

DELIVERY_LOCATION_ID_FK

DATEREQUIRED

*Data Type:*Date

Description:Specifies the date on which the requested container is to be delivered. The date is specified in the date-time literal format. For example, 01-27-2006.

Options:

*Required?:*N

Table Name:INV_REQUESTS

*Database Field Name:*DATE_REQUIRED

REQUESTCOMMENTS

*Data Type:*adVarChar

Description:Specifies the comments associated with the request.

Options:

Required?: N

Table Name:IINV_REQUESTS

Database Field Name:

REQUEST_COMMENTS

REQUESTTYPEID

*Data Type:*adNumeric

Description:Specifies the type of the request.

Options:

*Required?:*N

Table Name:IINV_REQUESTS

Database Field Name:

REQUEST_TYPE_ID_FK

CONTAINERTYPEID

*Data Type:*adNumeric

Description:Specifies the type of the sample containers.

Options:

*Required?:*Y

Table Name:INV_REQUESTS *Database Field Name:*

CONTAINER_TYPE_ID_FK

NUMCONTAINERS

*Data Type:*adNumeric

Description:Specifies the number of samples requested.

Options:

*Required?:*Y

Table Name:INV_REQUESTS

Database Field Name:

NUMBER_CONTAINERS

QTYLIST

*Data Type:*adVarChar

Description:Specifies the list of quantities that each sample container should contain.

Options:

*Required?:*Y

Table Name:INV_REQUESTS

*Database Field Name:*QUANTITY_LIST

SHIPTONAME

*Data Type:*adVarChar

Description:Specifies the user to whom the request is to be shipped.

Options:

*Required?:*N

Table Name:INV_REQUESTS

*Database Field Name:*SHIP_TO_NAME

EXPENSECENTER

*Data Type:*adVarChar

Description:Specifies the Expense Center for the sample request.

Options:

*Required?:*N

Table Name:INV_REQUESTS

*Database Field Name:*EXPENSE_CENTER

Output

The ID of the updated request, which is stored in the RequestID parameter.

Example

The following code updates a request with ID 121:

```
/cheminv/api/UpdateRequest.asp?RequestID=121&UserID=mjs&QtyRequired=10&LocationID=1123&ContainerTypeID=1&NumContainers=2&QtyList=10,10
```

UpdateReservation

Updates a container reservation.

Syntax

```
/cheminv/api/UpdateReservation.asp?ReservationID=adNumeric&ContainerID=adNumeric&ReservationUserID=adVarChar&QtyReserved=Int[&ReservationTypeID=adNumeric]
```

Inputs

RESERVATIONID

*Data Type:*adNumeric

Description:Specifies the ID of the reservation to be updated.

Options:

*Required?:*Y

Table Name:INV_RESERVATIONS

*Database Field Name:*RESERVATION_ID

CONTAINERID

*Data Type:*adNumeric

Description:Specifies the ID of the container the contents of which are to be reserved.

Options:

*Required?:*Y

Table Name:INV_RESERVATIONS

*Database Field Name:*CONTAINER_ID_FK

RESERVATIONUSERID

*Data Type:*adVarChar

Description:Specifies the ID of the user who wants to reserve the container contents.

Options:

*Required?:*Y

Table Name:INV_RESERVATIONS

*Database Field Name:*USER_ID_FK

QTYRESERVED

*Data Type:*adNumeric

Specifies the amount of the quantity to be reserved.

Options:

*Required?:*Y

Table Name:INV_RESERVATIONS

*Database Field Name:*QTY_RESERVED

RESERVATIONTYPEID

*Data Type:*adNumeric

Description:Specifies the ID of the type of reservation.

Options:

*Required?:*N

Table Name:INV_RESERVATIONS

Database Field Name: RESERVATION_TYPE_ID_FK

Output

The ID of the updated reservation, which is stored in the ReservationID parameter.

Example

The following code updates a reservation with ID 42:

```
/cheminv/api/UpdateReservation.asp?ReservationID=42&ContainerID=1742&ReservationUserID=INVADMIN&QtyReserved=.5
```

UpdateSubstance

Updates the attributes of an already existing substance.

Syntax

```
/cheminv/api/UpdateSubstance.asp?CompoundID=adNumeric&SubstanceName=adVarChar[&Structure=CLOB&CAS=adVarChar&ACX_ID=adVarChar&ALT_ID_1=adVarChar&ALT_ID_2=adVarChar&ALT_ID_3=adVarChar&ALT_ID_4=adVarChar&ALT_ID_5=adVarChar]
```

Inputs

COMPOUNDID

*Data Type:*adNumeric

Description:Specifies the ID of the compound to be updated.

Options:

*Required?:*Y

Table Name:INV_COMPOUNDS

*Database Field Name:*COMPOUND_ID

SUBSTANCENAME

*Data Type:*adVarChar

Description:Specifies the name of the compound.

Options:

*Required?:*Y

Table Name:INV_COMPOUNDS

*Database Field Name:*SUBSTANCE_NAME

STRUCTURE

*Data Type:*CLOB

Description:Specifies the chemical structure of the compound. The Structure parameter stores the chemical structure of the substance as the base64 encoded cdx data. Due to the length of cdx data, it is recommended to pass this data as the form data instead of passing it through the query string.

Options:

*Required?:*N

Table Name:INV_COMPOUNDS

*Database Field Name:*BASE64_CDX

CAS

*Data Type:*adVarChar

Description:Specifies the CAS number of the substance.

Options:

*Required?:*N

Table Name:INV_COMPOUNDS

*Database Field Name:*CAS

ACX_ID

*Data Type:*adVarChar

Description:Specifies the ACX ID of the substance.

Options:

*Required?:*N

Table Name:IINV_COMPOUNDS

*Database Field Name:*ACX_ID

ALT_ID_1

*Data Type:*adVarChar

Description:Specifies the value of the configurable field ALT_ID_1.

Options:

*Required?:*N

Table Name: INV_COMPOUNDS

*Database Field Name:*ALT_ID_1

ALT_ID_2

*Data Type:*adVarChar

Description:Specifies the value of the configurable field ALT_ID_2.

Options:

*Required?:*N

Table Name:INV_COMPOUNDS

*Database Field Name:*ALT_ID_2

ALT_ID_3

*Data Type:*adVarChar

Description:Specifies the value of the configurable field ALT_ID_3.

Options:

*Required?:*N

Table Name:INV_COMPOUNDS

*Database Field Name:*ALT_ID_3

ALT_ID_4

*Data Type:*adVarChar

Description:Specifies the value of the configurable field ALT_ID_4.

Options:

*Required?:*N

Table Name:INV_COMPOUNDS

*Database Field Name:*ALT_ID_4

ALT_ID_5

*Data Type:*adVarChar

Description:Specifies the value of the configurable field ALT_ID_5.

Options:

*Required?:*N

Table Name: INV_COMPOUNDS

*Database Field Name:*ALT_ID_5

Output

The ID of the updated compound. This ID is stored in the compound_id parameter, the data type of which is adNumeric.

Example

The following code updates a substance with ID 111:

```
cheminv/api/UpdateSubstance.asp?EditCompoundID=111&SubstanceName=Benzene[&CAS=71-43-2]
```

UpdateSynonym

Updates a synonym of a substance.

Syntax

```
/cheminv/api/UpdateSynonym.asp?SynonymID=adNumeric&SynonymName=adVarChar
```

Inputs

SYNONYMID

*Data Type:*adNumeric

Description:Specifies the ID of the synonym to be updated.

Options:

*Required?:*Y

Table Name:INV_SYNONYMS

*Database Field Name:*SYNONYM_ID

SYNONYMNAME

*Data Type:*adVarChar

Description:Specifies the updated synonym of the substance.

Options:

*Required?:*Y

Table Name:INV_SYNONYMS

*Database Field Name:*SUBSTANCE_NAME

Output

The ID of the updated synonym, which is stored in the SynonymID parameter.

Example

The following code updates a synonym with ID 121:

```
/cheminv/api/UpdateSynonym.asp?SynonymID=121&SynonymName="Benzenol"
```

UpdateTable

Updates the specified rows and columns of a table.

Syntax

```
/cheminv/api/UpdateTable.asp?TableName=adVarChar&pkColumnName=adVarChar&pkIDs=adVarChar&ValuePairs=adVarChar
```

Inputs

TABLENAME

*Data Type:*adVarChar

Description:Specifies the name of the table to be updated.

Options:

*Required?:*Y

Table Name:N/A

*Database Field Name:*N/A

PKCOLUMNNAME

*Data Type:*adVarChar

Description:Specifies a comma delimited list of the primary key column(s).

Options:

*Required?:*Y

Table Name:N/A

*Database Field Name:*N/A

PKIDS

*Data Type:*adVarChar

Description:Specifies a "::" delimited list of primary keys ID. *Options*:

*Required?:*Y

Table Name:N/A

*Database Field Name:*N/A

VALUEPAIRS

*Data Type:*adVarChar

Description:Specifies a "::" delimited list of the pairs of the name and value for the fields to be updated. For example: colnam1%3Dval1::colnam2%3Dval2.

Options:

*Required?:*Y

Table Name:N/A

*Database Field Name:*N/A

Output

None.

Example

The following code updates the INV_CONTAINERS table:

```
/cheminv/api/UpdateTable.asp?TableName=INV_CONTAINERS&pkColumnName=CONTAINER_ID&pkIDs=1497::1498&ValuePairs=LOCATION_ID_FK%3D1041::COMPOUND_ID_FK%3D447
```

UpdateWell

Updates the attributes of a well.

Syntax

```
/cheminv/api/UpdateWell.asp?PlateIDs=adVarChar&WellIDs=adVarChar&ValuePairs=adVarChar
```

Inputs

WELLIDS

*Data Type:*adVarChar

Description:Specifies a comma delimited list of the IDs of the wells to be updated.

Options:

*Required?:*Y

Table Name:INV_WELLS

*Database Field Name:*WELL_ID

PLATEIDS

*Data Type:*adVarChar

Description:Specifies a comma delimited list of the IDs of the plates, the wells associated with which are to be updated.

Options:

*Required?:*Y

Table Name:INV_WELLS

*Database Field Name:*PLATE_ID_FK

VALUEPAIRS

*Data Type:*adVarChar

Description:Specifies a "::" delimited list of the name and value pairs of the well fields to be updated. For example: colnam1%3Dval1::colnam2%3Dval2.

Options:

*Required?:*Y

Table Name:N/A

*Database Field Name:*N/A

Output

None.

Example

The following code updates the qty_remaining and quantity_unit_fk attribute of the A-1 and A-2 wells in the P111 plate:

```
/cheminv/api/UpdateWell.asp?PlateIDs=P111&WellIDs=A-1,A-2&ValuePairs=qty_remaining=10::quantity_unit_fk=2
```

UpdateWellContentType

Updates a well content type or well format.

Syntax

```
/cheminv/api/UpdateWellContentType.asp?WellFormatID=adNumeric&WellFormatName=adVarChar
```

Inputs

WELLFORMATID

*Data Type:*adNumeric

Description:Specifies the ID of the well format to be updated.

Options:

*Required?:*Y

Table Name:INV_ENUMERATION

*Database Field Name:*ENUM_ID

WELLFORMATNAME

*Data Type:*adVarChar

Description:Specifies the name of the well format.

Options:

*Required?:*Y

Table Name:INV_ENUMERATION

*Database Field Name:*ENUM_VALUE

Output

The ID of the updated well format. This ID is stored in the WellFormatID parameter.

Example

The following code updates a well format with ID 6:

```
/cheminv/api/UpdateWellContent-
Type.asp?WellFormatID=6&WellFormat-
Name=NewWellFormat
```

BioSar API

ChemOffice WebServer applications are typically driven from dynamically generated web pages created by the server itself. That is, a user can query the server by entering search parameters into an HTML form hosted on the web server, and search results are passed back to the browser as HTML. However, the ChemOffice WebServer also supports remote requests from applications capable of sending/receiving HTTP requests/ responses. This allows for external applications to initiate searches on the server and to take on the responsibility of post-processing and/or displaying the search results. The following is a list of remote requests to the server:

Search

Get_Structure

Query_String

Search

The Search dataaction initiates a search over the application's database/s. The results of the search are stored in the CSDOHitList table within the application's database. The hitlist table will contain ID values corresponding to the primary key of the base table as defined in the formgroup.

NOTE: All parameters can be passed either as Form (POST) or Querystring (GET) data.

Syntax

<ServerName>/<ApplicationName>/<ApplicationName>/<ApplicationName>_action.asp?formgroup=Text&dbname=Text&DataAction=Search[&return_location=Text&return_data=Text&
store_request=Boolean&metadata_directive=Text&reload_basers=Boolean]

Inputs

formgroup = <formgroupName>

The name of a valid formgroup as defined in the applications ini file, or dynamically defined via the metadata_directive options.

dbname = <DatabaseName>

The name of the dataview ini file for the database being accessed.
Possible options : reg (for Registration System), cheminv (for Inventory Manager), chemacx, wdi, wda
Example: dbname= chemacx

[return_location] = Empty|<ReturnURL>

The URL where the server will redirect or transfer the request after the search is completed.

By default the return location is determined by the formgroup settings in the application ini file. This parameter can be filled with a URL on the applications server specified relative to the location of the action page.

[return_data] = Empty|Redirect|Transfer|CSV|HitlistID

Specifies how or what data should be returned by the server.

By default the data is returned by causing a client side redirection to the results page defined in the formgroup. The Redirect option forces a server side redirect (HTTP 302) to the URL specified by the return location. The Transfer option forces a server process transfer to the URL specified in the return location. The transfer process preserves the initial asp request object for post processing in the results page.

The CSV option causes the server to respond with the hitlist in a comma separated values list. The HitlistID option causes the server to return a single HitlistId value which can be used to retrieve the hitlist from the database table during post-processing .

[store_request] = Empty|True

If set to true, the server stores the initial query request as a data dictionary in session memory. Query parameters can be accessed from the results page by interrogating one of two scripting dictionary objects stored in Session("Request_FormObject" & dbkey) and Session("Request_QueryObject" & dbkey).

NOTE: that setting return_data=transfer is a more efficient way to pass the request object to the results page. However, the transfer option is only available in Win2k/IIS5.0 server configurations.

[metadata_directive] = Empty|bypass_ini|blind

Controls where the server obtains the meta data it requires to dynamically build ChemSQL commands from the posted query fields. If omitted , the server uses a combination of the meta data posted by the form and meta data from the application ini file. The bypass_ini file is not fully implemented at this point, but it will allow the query page to dynamically provide all required meta data. The blind option forces the server to use ini data exclusively thus simplifying the query specification.

[reload_basers] = Empty|True

If set to true, causes the server to discard the first section of the hitlist returned by the CSDO search, and to reload it from the CSDOHitlist table. This allows for post-processing of the CSDOhitlist values before results are displayed.

Get_Structure

Retrieves a chemical structure from a ChemOffice WebServer database and returns structure data in the requested format.

NOTE: All parameters can be passed either as Form (POST) or Querystring (GET) data.

Syntax

<ServerName>/<ApplicationName>/<ApplicationName>/<ApplicationName>_action.asp?dbname=adVarChar&dataaction=get_structure&formgroup=adVarChar&Table=adVarChar&Field=adVarChar&
DisplayType=adVarChar&StrucID=adNumeric[&width=adNumeric&height=adNumeric]

Inputs

dbName= <DatabaseName>

The name of the dataview ini file for the database being accessed.

Possible options : reg (for Registration System), cheminv (for Inventory Manager), chemacx, wdi, wda
Example: dbname= chemacx

dataaction = get_structure

Must be set equal to "get_structure".

formgroup = <formgroupName>

The name of a valid formgroup as defined in the applications ini file, or dynamically defined via the metadata_directive options.

Table= <table_name>

The name of the table containing the unique identifier associated with the structure to be fetched.

NOTE: A valid ChemConnection to that table must exists in the ini file

Example: Table = Substance

Field = <field_name>

The name of the structure field in the associated ChemFinder form.
The fully qualified name of a structure field (this field has a Base64cdx datatype).

NOTE: The field name must match the structure field name on the ChemFinder form.

Example: Field = structure

DisplayType= cdx|gif|sizedgif

Determines the file type to be used when saving the structure. When equal to sizedgif, a gif is returned.

Example: DisplayType = cdx

StrucID= <ID>

The unique numerical identifier (MOL_ID) of the chemical structure.
Example: StrucID = 6

[width]=<gif_width>

The width of the gif being displayed. This is an optional parameter used when DisplayType= sizedgif. The default is 200.

[height]=<gif_height>

The height of the gif being displayed. This is an optional parameter used when DisplayType= sizedgif. The default is 200.

Query_string

Allows for a text based query to be initiated via an HTTP GET request from a page or application external to the ChemOffice WebServer. The results are returned in HTML as if the search had been initiated from the standard query page.

NOTE: In this instance the page being requested is default.asp rather than apname_action.asp. All parameters can be passed either as Form (POST) or Querystring (GET) data.

Syntax

<ServerName>/<ApplicationName>/<ApplicationName>/default.asp?dbname=adVarChar&dataaction=query_string&formgroup=adVarChar&field_type=adVarChar&full_field_name=adVarChar&field_value=adVarChar|adNumeric

Inputs

dbName= <DatabaseName>

The name of the dataview ini file for the database being accessed.
Possible options : reg (for Registration System), cheminv (for Inventory Manager), chemacx, wdi, wda
Example: dbname= chemacx

dataaction = query_string

Must be set equal to "query_string".

formgroup = <formgroupName>

The name of a valid formgroup as defined in the applications ini file, or dynamically defined via the metadata_directive options.

field_type= TEXT|INTEGER

Defines the data type of the text field being used to query the database.

Example: field_type= TEXT

full_field_name = <field name>

The fully qualified name of the field to be used in the query.

NOTE: The table and must field must be properly declared in the ini file for the application

Example: full_field_name= Substance.CAS

field_value= <value>

The value of the field to search on.
Example: field_value = 50-50-0.

Registration System API

This Includes the **ChemOffice Webserver API**:

Search

Get_Structure

Query_String

Search

The Search dataaction initiates a search over the application's database/s. The results of the search are stored in the CSDOHitList table within the application's database. The hitlist table will contain ID values corresponding to the primary key of the base table as defined in the formgroup.

NOTE: All parameters can be passed either as Form (POST) or Querystring (GET) data.

Syntax

<ServerName>/<ApplicationName>/<ApplicationName>/<ApplicationName>_action.asp?formgroup=Text&dbname=Text&DataAction=Search[&return_location=Text&return_data=Text&store_request=Boolean&metadata_directive=Text&reload_basers=Boolean]

Inputs

formgroup = <formgroupName>

The name of a valid formgroup as defined in the applications ini file, or dynamically defined via the metadata_directive options.

dbname = <DatabaseName>

The name of the dataview ini file for the database being accessed.
Possible options : reg (for Registration System), cheminv (for Inventory Manager), chemacx, wdi, wda
Example: dbname= chemacx

[return_location] = Empty|<ReturnURL>

The URL where the server will redirect or transfer the request after the search is completed.

By default the return location is determined by the formgroup settings in the application ini file. This parameter can be filled with a URL on the applications server specified relative to the location of the action page.

[return_data] = Empty|Redirect|Transfer|CSV|HitlistID

Specifies how or what data should be returned by the server.

By default the data is returned by causing a client side redirection to the results page defined in the formgroup. The Redirect option forces a server side redirect (HTTP 302) to the URL specified by the return location. The Transfer option forces a server process transfer to the URL specified in the return location. The transfer process preserves the initial asp request object for post processing in the results page.

The CSV option causes the server to respond with the hitlist in a comma separated values list. The HitlistID option causes the server to return a single HitlistId value which can be used to retrieve the hitlist from the database table during post-processing .

[store_request] = Empty|True

If set to true, the server stores the initial query request as a data dictionary in session memory. Query parameters can be accessed from the results page by interrogating one of two scripting dictionary objects stored in Session("Request_FormObject" & dbkey) and Session("Request_QueryObject" & dbkey).

Note that setting return_data=transfer is a more efficient way to pass the request object to the results page. However, the transfer option is only available in Win2k/IIS5.0 server configurations.

[metadata_directive] = Empty|bypass_ini|blind

Controls where the server obtains the meta data it requires to dynamically build ChemSQL commands from the posted query fields. If omitted , the server uses a combination of the meta data posted by the form and meta data from the application ini file. The bypass_ini file is not fully implemented at this point, but it will allow the query page to dynamically provide all required meta data. The blind option forces the server to use ini data exclusively thus simplifying the query specification.

[reload_basers] = Empty|True

If set to true, causes the server to discard the first section of the hitlist returned by the CSDO search, and to reload it from the CSDOHitlist table. This allows for post-processing of the CSDOhitlist values before results are displayed.

Get_structure

Retrieves a chemical structure from a ChemOffice WebServer database and returns structure data in the requested format.

NOTE: All parameters can be passed either as Form (POST) or Querystring (GET) data.

Syntax

```
<ServerName>/<ApplicationName>/<ApplicationName>/<ApplicationName>_action.asp?dbname=adVarChar&dataaction=get_structure&formgroup=adVarChar&Table=adVarChar&Field=adVarChar&DisplayType=adVarChar&StrucID=adNumeric[&width=adNumeric&height=adNumeric]
```

Inputs

dbName= <DatabaseName>

The name of the dataview ini file for the database being accessed.
Possible options : reg (for Registration System), cheminv (for Inventory Manager), chemacx, wdi, wda
Example: dbname= chemacx

dataaction = get_structure

Must be set equal to "get_structure".

formgroup = <formgroupName>

The name of a valid formgroup as defined in the applications ini file, or dynamically defined via the metadata_directive options.

Table= <table_name>

The name of the table containing the unique identifier associated with the structure to be fetched.

NOTE: A valid ChemConnection to that table must exists in the ini file

Example: Table = Substance

Field = <field_name>

The name of the structure field in the associated ChemFinder form.
The fully qualified name of a structure field (this field has a Base64cdx datatype).

NOTE: Note: The field name must match the structure field name on the ChemFinder form.

Example: Field = structure

DisplayType= cdx|gif|sizedgif

Determines the file type to be used when saving the structure. When equal to sizedgif, a gif is returned.
Example: DisplayType = cdx

StrucID= <ID>

The unique numerical identifier (MOL_ID) of the chemical structure.
Example: StrucID = 6

[width]=<gif_width>

The width of the gif being displayed. This is an optional parameter used when DisplayType= sizedgif. The default is 200.

[height]=<gif_height>

The height of the gif being displayed. This is an optional parameter used when DisplayType= sizedgif. The default is 200.

Query_string

Allows for a text based query to be initiated via an HTTP GET request from a page or application external to the ChemOffice WebServer. The results are returned in HTML as if the search had been initiated from the standard query page.

NOTE: Note: In this instance the page being requested is default.asp rather than apname_action.asp. All parameters can be passed either as Form (POST) or Querystring (GET) data.

Syntax

<ServerName>/<ApplicationName>/<ApplicationName>/default.asp?dbname=adVarChar&dataaction=query_string&formgroup=adVarChar&field_type=adVarChar&full_field_name=adVarChar&field_value=adVarChar|adNumeric

Inputs

dbName= <DatabaseName>

The name of the dataview ini file for the database being accessed.
Possible options : reg (for Registration System), cheminv (for Inventory Manager), chemacx, wdi, wda
Example: dbname= chemacx

dataaction = query_string

Must be set equal to "query_string".

formgroup = <formgroupName>

The name of a valid formgroup as defined in the applications ini file, or dynamically defined via the metadata_directive options.

field_type= TEXT|INTEGER

Defines the data type of the text field being used to query the database.
Example: field_type= TEXT

full_field_name = <field name>

The fully qualified name of the field to be used in the query.

NOTE: The table and must field must be properly declared in the ini file for the application

Example: full_field_name= Substance.CAS

field_value= <value>

The value of the field to search on.
Example: field_value = 50-50-0

Other Registration API

Registration System includes a number of routines available to the user to manipulate data within the database. These functions are called the Chemical Registration HTTP API. The following is a complete list of functions, their purpose, and how to use them:

Search

Reg_temp

Reg_perm

DB_info

Search

The Search postaction initiates a search over the Registration System database.

NOTE: In general, the POST method should be used for requests that contain BASE64_CDX. Be sure to URL encode all BASE64_CDX requests. Remove all carraige returns.

Syntax

```
<ServerName>/reg/reg_post_action.asp?reg_method=search&user_id=adVarChar&user_pwd=adVarChar&<Field>=adVarChar&struc_search_type=adVarChar&return_fields=adVarChar[&return_structure=adVarChar&return_format=adVarChar&return_embedded=adVarChar]
```

Inputs

reg_method = Search

Must be equal to Search.

USER_ID = <User ID>

The ID of a valid Registration System User. The User must have the correct permissions for the requested task.
Example: USER_ID= T5_84

USER_PWD = <User Password>

The password for the username already entered. The User must have the correct permissions for the requested task.
Example: USER_PWD= T5_84

<Field>= <structure>

Where <Field> is the name of a field whose value should be the Base64 string created with the ChemDraw plugin and <structure> is the Base64 string.
Example: reg_numbers.Structure= <Base64 string>

STRUC_SEARCH_TYPE = EXACT|Substructure

The Search Type used for the structure entered. The default is EXACT.
Example: STRUC_SEARCH_TYPE= EXACT

[RETURN_FIELDS] = <field names>

The names of fields whose value should be returned. This can be any field, but it must be a field found in the .ini file. The default is the primary key for the base table. Filed names should be separated by a comma.
Example: RETURN_FIELDS= Reg_Numbers.Reg_Number

[RETURN_STRUCTURE] = TRUE|FALSE

When set to TRUE, specifies that a structure is to be returned. Default value is FALSE.

NOTE: Return_format and return_embedded are required parameters if set to TRUE.

[RETURN_FORMAT] = CDX|GIF

The return format desired. The default is CDX.

NOTE: only used when return_structure is set to TRUE

[RETURN_EMBEDDED] = TRUE|FALSE

> For *GIF* format: If TRUE then the the GIF is returned in an <IMG></IMG> tag and is thus displayed as a structure - if FALSE only a file path to the gif is returned.
> For *CDX* format: If TRUE the structure is returned within and <EMBED></EMBED> and displayed using the plugin.

The default value is TRUE.

NOTE: This field is only used when return_structure is set to TRUE

Reg_temp

Registers a temporary compound, batch, salt, or identifier to the Temporary database.

This should be sent as an HTTP POST request. In general, the POST method should be used for requests that contain BASE64_CDX

Syntax

```
.<ServerName>/reg/
reg_post_action.asp?reg_method=reg_temp
&user_id=adVarChar&
user_pwd=adVarChar&reg_parameter=ad-
Var-
Char&[Temporary_Structures.cpd_internal_i
d=adVar-
Char&Temporary_Structures.reg_id=adVar-
Char&Temporary_Structures.Project_id=ad
Var-

Char&Temporary_Structures.Compound_Ty
pe=adVar-
Char&Temporary_Structures.Notebook_Tex
t=adVar-
Char&Temporary_Structures.Salt_Code=ad-
VarChar&Temporary_Structures.Structure=a
dVarChar&full_field_name=adVarChar]
```

Inputs

reg_method = reg_temp

Must be equal to reg_temp.

USER_ID = <User ID>

The ID of a valid Chemical Registration User. The User must have the correct permissions for the requested task.

Example: USER_ID=T5_84

USER_PWD = <User Password>

The password for the username already entered. The User must have the correct permissions for the requested task.

Example: USER_PWD=T5_84

reg_parameter= ADD_COMPOUND|ADD_BATCH|ADD_SALT| ADD|IDENTIFIER|AUTHENTICATE

Defaults to ADD_COMPOUND if a structure is sent, ADD_BATCH if there is no structure sent.

Depending on the reg_parameter passed, some of the following optional parameters may be required.

[Temporary_Structures.cpd_internal_id] = <ID>

The CPD ID of the substance being added. This is a required field for a reg_parameter of ADD_SALT, and is automatically filled if left empty.

[Temporary_Structures.reg_id] = <ID>

The Registration ID of the substance being added to. This is a required field for a reg_parameter of ADD_BATCH and ADD_IDENTIFIER, and is automatically filled if left empty.

[Temporary_Structures.Scientist_id] = <ID>

The Scientist ID number for the substance being added. This is automatically filled if left empty.

[Temporary_Structures.Sequence_id] = <ID>

The Sequence ID number for the substance being added. This is automatically filled if left empty.

[Temporary_Structures.Project_id] = <ID>

The Project ID number for the substance being added. This is automatically filled if left empty.

[Temporary_Structures.Compound_Type] = <compound type>

The Compound Type ID number for the substance being added. This is automatically filled if left empty.

[Temporary_Structures.Notebook_Text] = <text>

The notebook text.This is automatically filled if left empty.

[Temporary_Structures.Salt_Code] = <code>

The Salt Code for the salt being added. This is automatically filled if left empty

[Temporary_Structures.Structure] = <base64 string>

The Base64 string output from the ChemDraw plugin reprsenting the structure drawn. This is a required field for a reg_parameter of ADD_COMPOUND.

[full_field_name] = <field value>

The fully qualified name of the field to be used in the query equal to the value. Any field vaue pair can be used in this query.

Example: Temporary_Structures.Compound_Name=benzene

Reg_perm

Registers a compound, batch, salt, or identifier to the Permanent database.

This should be sent as an HTTP POST re-

quest. In general, the POST method should be used for requests that contain BASE64_CDX.

SYNTAX

<ServerName>/reg/reg_post_action.asp?reg_method=reg_temp&user_id=adVarChar&user_pwd=adVarChar&reg_parameter=adVarChar&[Temporary_Structures.temp_compound_id=adVarChar&Temporary_Structures.cpd_internal_id&Temporary_Structures.reg_id&Temporary_Structures.Scientist_id=adVarChar&Temporary_Structures.Sequence_id=adVarChar&Temporary_Structures.Project_id=adVarChar&Temporary_Structures.Compound_Type=adVarChar&Temporary_Structures.Notebook_Text=adVarChar&Temporary_Structures.Salt_Code=adVarChar&Temporary_Structures.Structure=adVarChar&full_field_name=adVarChar

INPUTS

reg_method = reg_perm

Must be equal to reg_perm.

USER_ID = <User ID>

The ID of a valid Chemical Registration User. The User must have the correct permissions for the requested task.

Example: USER_ID=T5_84

USER_PWD = <User Password>

The password for the username already entered. The User must have the correct permissions for the requested task.

Example: USER_PWD=T5_84

reg_parameter= OVERRIDE|UNIQUE|UNIQUE_DEL_TEMP2|USER_INPUT|NEW_BATCH|NEW_SALT|AUTHENTICATE

Defaults to UNIQUE_DEL_TEMP2

reg_parameter	**Description**
OVERRIDE	adds a new compound even if dups are found
UNIQUE	adds only if no duplicates are found - leaves item in temporary table
UNIQUE_DEL_TEMP2	adds only if no duplicates are found and removes from temp table
USER_INPUT	returns duplicate_ids and temp_compound_id for user decision on dups action
NEW_BATCH	adds a new batch if for the temp_compound_id sent with the request
AUTHENTICATE	looks to see if the user has rights to perform the action and sends a valid response if unable

Depending on the reg_parameter passed, some of the following optional parametres may be required.(See “Reg_perm Required Fields”)

Reg_perm Required Fields

The following is a list of additional fields required to use reg_perm.

reg_parameters	Required Fields
OVERRIDE	all fields required
UNIQUE	all fields required
UNIQUE_DEL_TEMP	all fields required
USER_INPUT	all fields required
NEW_BATCH	Temporary_Structures.reg_id, Temporary_Structures.temp_compound_id
NEW_SALT	Temporary_Structures.reg_id, Temporary_Structures.temp_compound_id

[Temporary_Structures.temp_compound_id] = <ID>

The compound ID for the substance in the temporary database. This is a required field for a reg_parameter of NEW_SALT, and is automatically filled if left empty.

[Temporary_Structures.cpd_internal_id] = <ID>

The CPD ID of the substance being added. This is a required field for a reg_parameter of NEW_SALT, and is automatically filled if left empty.

[Temporary_Structures.reg_id] = <ID>

The Registration ID of the substance being added to. This is a required field for a reg_parameter of NEW_BATCH.

[Temporary_Structures.Scientist_id] = <ID>

The Scientist ID number for the substance being added. This is automatically filled if left empty.

[Temporary_Structures.Sequence_id] = <ID>

The Sequence ID number for the substance being added. This is automatically filled if left empty.

[Temporary_Structures.Project_id] = <ID>

The Project ID number for the substance being added. This is automatically filled if left empty.

[Temporary_Structures.Compound_Type] = <compound type>

The Compound Type ID number for the substance being added. This is automatically filled if left empty.

[Temporary_Structures.Notebook_Text] = <text>

The notebook text.This is automatically filled if left empty.

[Temporary_Structures.Salt_Code] = <code>

The Salt Code for the salt being added. This is automatically filled if left empty

[Temporary_Structures.Structure] = <base64 string>

The Base64 string output from the ChemDraw plugin reprsenting the structure drawn. This is a required field for a reg_parameter of ADD_COMPOUND.

[full_field_name] = <field value>

The fully qualified name of the field to be used in the query equal to the value. Any field vaue pair can be used in this query.

Example:
Temporary_Structures.Compound_Name=benzene

Db_info

Allows the user to retrieve information about oracle database via select statements.

NOTE: In general, the POST method should be used for requests that contain

---

BASE64_CDX. Be sure to URL encode all BASE64_CDX requests. Remove all carraige returns.

Syntax

<ServerName>/reg/reg_post_action.asp?reg_method=db_info&user_id=adVarChar&user_pwd=adVarChar&SQL_request=adVarChar&return_props=adVarChar

Inputs

reg_method = db_info

Must be equal to db_info.

reg_parameter = Empty|AUTHENTICATE

Use AUTHENTICATE if you want check user_id/userpassword. Otherwise, leave blank.

USER_ID = <User ID>

The ID of a valid Registration System User. The User must have the correct permissions for the requested task.
Example: USER_ID= T5_84

USER_PWD = <User Password>

The password for the username already entered. The User must have the correct permissions for the requested task.
Example: USER_PWD= T5_84

SQL_REQUEST= <select statement>

A valid SQL statement. UPDATE, DELETE, AND INSERT statements not supported. This is an INFO only method

RETURN_PROPS= Empty|name|numericscale|precision|size|type|value

The properties of each field to return. Should be entered as a comma delimited list. If left empty, defaults to value.

Outputs

Returns info from select statement as indicated in return_props. If return_props is empty then the value is returned by default.

Example

To retrieve information about the fields in each of the tables found from select * from tab, use the following (shown here for ALT_IDS):

```
select column_name,data_type from
user_tab_columns where table_name =
'ALT_IDS'
```

This is information posted when you click the Submit button: Post Action and Post name/value pairs information (in concatenated form):

```
/reg/reg_post_action.asp?reg_method=dbinfo&user_id=regdb&user_pwd=oracle&sql_request
```

```
=select* from
tab&return_props=name,numerics-
cale,precision,size,type,value
```

ChemACX API

This Includes the ChemOffice Enterprise API:

Search

Get_Structure

Query_String

Search

The Search dataaction initiates a search over the application's database/s. The results of the search are stored in the CSDOHitList table within the application's database. The hitlist table will contain ID values corresponding to the primary key of the base table as defined in the formgroup.

> *NOTE: All parameters can be passed either as Form (POST) or Querystring (GET) data.*

Syntax

```
<ServerName>/<ApplicationName>/<ApplicationName>/<ApplicationName>_action.asp?formgroup=Text&dbname=Text&DataAction=Search[&return_location=Text&return_data=Text&store_request=Boolean&metadata_directive=Text&reload_basers=Boolean]
```

Inputs

formgroup = <formgroupName>

The name of a valid formgroup as defined in the applications ini file, or dynamically defined via the metadata_directive options.

dbname = <DatabaseName>

The name of the dataview ini file for the database being accessed.
Possible options : reg (for Registration System), cheminv (for Inventory Manager), chemacx, wdi, wda
Example: dbname= chemacx

[return_location] = Empty|<ReturnURL>

The URL where the server will redirect or transfer the request after the search is completed.

By default the return location is determined by the formgroup settings in the application ini file. This parameter can be filled with a URL on the applications server specified relative to the location of the action page.

[return_data] = Empty|Redirect|Transfer|CSV|HitlistID

Specifies how or what data should be returned by the server.

By default the data is returned by causing a client side redirection to the results page defined in the formgroup. The Redirect option forces a server side redirect (HTTP 302) to the URL specified by the return location. The Transfer option forces a server process transfer to the URL specified in the return location. The transfer process preserves the initial asp request object for post processing in the results page.

The CSV option causes the server to respond with the hitlist in a comma separated values list. The HitlistID option causes the server to return a single HitlistId value which can be used to retrieve the hitlist from the database table during post-processing .

[store_request] = Empty|True

If set to true, the server stores the initial query request as a data dictionary in session memory. Query parameters can be accessed from the results page by interrogating one of two scripting dictionary objects stored in Session("Request_FormObject" & dbkey) and Session("Request_QueryObject" & dbkey).

Note that setting return_data=transfer is a more efficient way to pass the request object to the results page. However, the transfer option is only available in Win2k/IIS5.0 server configurations.

[metadata_directive] = Empty|bypass_ini|blind

Controls where the server obtains the meta data it requires to dynamically build ChemSQL commands from the posted query fields. If omitted , the server uses a combination of the meta data posted by the form and meta data from the application ini file. The bypass_ini file is not fully implemented at this point, but it will allow the query page to dynamically provide all required meta data. The blind option forces the server to use ini data exclusively thus simplifying the query specification.

[reload_basers] = Empty|True

If set to true, causes the server to discard the first section of the hitlist returned by the CSDO search, and to reload it from the CSDOHitlist table. This allows for post-processing of the CSDOhitlist values before results are displayed.

Get_structure

Retrieves a chemical structure from a ChemOffice WebServer database and returns structure data in the requested format.

NOTE: All parameters can be passed either as Form (POST) or Querystring (GET) data.

Syntax

```
<ServerName>/<ApplicationName>/<ApplicationName>/<ApplicationName>_action.asp?dbname=adVarChar&dataaction=get_structure&formgroup=adVarChar&Table=adVarChar&Field=adVarChar &
DisplayType=adVarChar&StrucID=adNumeric[&width=adNumeric&height=adNumeric]
```

Inputs

dbName= <DatabaseName>

The name of the dataview ini file for the database being accessed.
Possible options : reg (for Registration System), cheminv (for Inventory Manager), chemacx, wdi, wda
Example: dbname= chemacx

dataaction = get_structure

Must be set equal to "get_structure".

formgroup = <formgroupName>

The name of a valid formgroup as defined in the applications ini file, or dynamically defined via the metadata_directive options.

Table= <table_name>

The name of the table containing the unique identifier associated with the structure to be fetched.
Note: A valid ChemConnection to that table must exists in the ini file
Example: Table = Substance

Field = <field_name>

The name of the structure field in the associated ChemFinder form.
The fully qualified name of a structure field (this field has a Base64cdx datatype).

NOTE: The field name must match the structure field name on the ChemFinder form.

Example: Field = structure

DisplayType= cdx|gif|sizedgif

Determines the file type to be used when saving the structure. When equal to sizedgif, a gif is returned.
Example: DisplayType = cdx

StrucID= <ID>

The unique numerical identifier (MOL_ID) of the chemical structure.
Example: StrucID = 6

[width]=<gif_width>

The width of the gif being displayed. This is an optional parameter used when DisplayType= sizedgif. The default is 200.

[height]=<gif_height>

The height of the gif being displayed. This is an optional parameter used when DisplayType= sizedgif. The default is 200.

Query_string

Allows for a text based query to be initiated via an HTTP GET request from a page or application external to the ChemOffice WebServer. The results are returned in HTML as if the search had been initiated from the standard query page.

NOTE: In this instance the page being requested is default.asp rather than apname_action.asp. All parameters can be passed either as Form (POST) or Querystring (GET) data.

Syntax

<ServerName>/<ApplicationName>/<ApplicationName>/default.asp?dbname=adVarChar&dataaction=query_string&formgroup=adVarChar&field_type=adVarChar&full_field_name=adVarChar&field_value=adVarChar|adNumeric

Inputs

dbName= <DatabaseName>

The name of the dataview ini file for the database being accessed.
Possible options : reg (for Registration System), cheminv (for Inventory Manager), chemacx, wdi, wda
Example: dbname= chemacx

dataaction = query_string

Must be set equal to "query_string".

formgroup = <formgroupName>

The name of a valid formgroup as defined in the applications ini file, or dynamically defined via the metadata_directive options.

field_type= TEXT|INTEGER

Defines the data type of the text field being used to query the database.
Example: field_type= TEXT

full_field_name = <field name>

The fully qualified name of the field to be used in the query.

NOTE: The table and must field must be properly declared in the ini file for the application

Example: full_field_name= Substance.CAS

field_value= <value>

The value of the field to search on.
Example: field_value = 50-50-0

Some other **ChemACX API References** are:

DisplayACXStruc

GetXMLdata

DisplayACXStruc

Returns a ChemACX structure given a StrucID. The structure can be returned in CDX or GIF format.

Syntax

```
/chemacx/api/DisplayACX-
Struc.asp?StrucID=adNumeric[&gif-
Width=adNumeric&gifHeight=adNumeric]
```

StrucID=<structureID>

The unique numerical identifier (MOL_ID) of the chemical structure.

Example: StrucID = 6

If this is the only parameter supplied, the structure will be returned in CDX format.

[gifWidth]= <width>

The width of the gif being displayed. This is an optional parameter used if you want to return the structure as a GIF instead of CDX.

[gifHeight]=<height>

The height of the gif being displayed. This is an optional parameter used if you want to return the structure as a GIF instead of CDX.

GetXMLdata

Given a list of CsNums, Product IDs, Package IDs, CAS numbers, or ACX IDs, returns supplier information in XML format.

Syntax

```
/chemacx/api/GetXMLdata.asp?field-
Name=adVarChar&valueL-
ist=CSV&sStructType=adVarChar[&synon
ym=adBoolean&product=adBoolean&prod-
Property=adBoolean&package=adBool-
ean&supplier=adBoolean&supplierAddre
ss=adBoolean&supplierEmail=adBool-
ean&supplierURL=adBoolean&supplier-
Phone=adBoolean]
```

fieldName=CsNum|ProductID|PackageID|CAS|ACX_ID

The Oracle field's name for which a list of values is being supplied. This field is the only search field available with the use of this page.

valueList=<listof values>

A list of values for the field name entered in fieldName. The list should be comma separated.

sStructType=cdx|gif|base64cdx|mol|Empty

The format of the structure data used in the datagram.

[synonym]=1|Empty

Equal to 1 if you would like synonyms returned. Defaults to 0 (no synonyms returned).

[product]=1|Empty

Equal to 1 if you would like product names returned. Defaults to 0 (no product names returned).

[prodProperty]=1|Empty

Equal to 1 if you would like product properties returned. Defaults to 0 (no product properties returned).

[package]=1|Empty

Equal to 1 if you would like packages returned. Defaults to 0 (no packages returned).

[supplier]=1|Empty

Equal to 1 if you would like supplier names returned. Defaults to 0 (no supplier names returned).

[supplierAddress]=1|Empty

Equal to 1 if you would like supplier addresses returned. Defaults to 0 (no supplier addresses returned).

[supplierEmail]=1|Empty

Equal to 1 if you would like supplier email addresses returned. Defaults to 0 (no supplier email addresses returned).

[supplierURL]=1|Empty

Equal to 1 if you would like supplier URLs returned. Defaults to 0 (no supplier URLs returned).

[supplierPhone]=1|Empty

Equal to 1 if you would like supplier phone numbers returned. Defaults to 0 (no supplier phone numbers returned).

Index