

RfIP Web Service API

Ver. 1.6

3/9/2007

RfIP Web Service API	1
1 Summary.....	7
2 Asset Management.....	8
2.1 GetAsset	8
2.2 GetNumAssets.....	8
2.3 GetAssetsInCategory.....	8
2.4 GetAssetsInDepartment	8
2.5 GetAssetByInventoryID	9
2.6 DeleteAsset.....	9
2.7 AddAsset	9
2.8 UpdateAsset.....	9
2.9 RemoveAssetFromDepartment.....	10
2.10 GetAssetForTag.....	10
2.11 GetAssetLocation	10
2.12 CleanUpDeletedAssets.....	10
2.13 GetAssetAttribute.....	11
2.14 GetAttributesForAsset	11
2.15 UpdateAssetAttribute	11
3 Category Management.....	12
3.1 GetCategory.....	12
3.2 GetNumCategories.....	12
3.3 GetNumberOfAssetsInCategory.....	12
3.4 DeleteCategory.....	12
3.5 AddCategory	12
3.6 UpdateCategory.....	13
3.7 AddCategoryAttribute.....	13
3.8 DeleteCategoryAttribute	13
3.9 GetCategoryAttribute	14
3.10 GetAttributesForCategory	14
3.11 GetInheritedAttributesForCategory	14
4 Department Management	15
4.1 GetDepartment.....	15
4.2 GetDepartments	15
4.3 GetNumDepartments	15
4.4 GetDepartmentByName.....	15
4.5 DeleteDepartment	15
4.6 AddDepartment	16
4.7 UpdateDepartment	16
4.8 DeleteDepartmentWorldLocationMapping.....	16
4.9 AddDepartmentWorldLocationMapping	16
4.10 GetWorldLocationMappingsForDepartment.....	17
5 Location Image Management	18
5.1 GetLocationImage	18

5.2	GetLocationMappingsImage.....	18
5.3	GetImageIDForLocation	18
5.4	GetImageCoordinatesForLocation.....	18
6	Location History Management	20
6.1	GetAssetHistoryCount	20
6.2	GetRawTagHistoryAtLocation	20
6.3	GetRawTagHistory	20
6.4	GetLocationsVisibleByTag	21
6.5	GetTagsVisibleAtLocation.....	21
6.6	GetAssetHistory	21
6.7	GetLastKnownTagPosition	22
6.8	GetTagLocation	22
6.9	GetLocationOfAssets	22
6.10	GetLastLocationOfMissingAssets.....	23
6.11	GetInventoryAtLocation.....	23
7	Virtual Location Management	25
7.1	GetLocation.....	25
7.2	DeleteLocation	25
7.3	GetLocations	25
7.4	AddLocation	25
7.5	UpdateLocation	26
7.6	GetLocationByPath.....	26
7.7	GetUnboundLocations	27
7.8	GetLocationsForDevice	27
8	Log Management	28
8.1	GetNumberOfRfIPDeviceLogEntries	28
8.2	GetNumberOfMonarchLogEntries	28
8.3	LogRfIPDeviceMessage	28
8.4	LogMonarchMessage.....	29
8.5	PurgeMonarchLogEntries.....	29
8.6	PurgeRfIPDeviceLogEntries.....	29
8.7	GetRfIPDeviceLogEntries	30
8.8	GetRfIPDeviceLogEntriesByPage	30
8.9	GetMonarchLogEntries	30
8.10	GetMonarchLogEntriesByPage.....	31
9	RFID Reader Management	32
9.1	GetRFIDReaders	32
9.2	GetRFIDReader	32
9.3	DeleteRFIDReader	32
9.4	AddRFIDReader	32
9.5	UpdateRFIDReader	33
9.6	GetRFIDReaderManufacturers.....	33
9.7	GetRFIDReaderModelsForManufacturer	33

9.8	GetRFIDReaderManufacturerForModel	33
9.9	GetRFIDReaderModel	34
9.10	GetRFIDReaderStatus	34
9.11	TestReaderConnectivity	34
9.12	ValidateReaderConfigurationParameters	34
9.13	StartRFIDReader	35
9.14	StopRFIDReader	35
10	RfIP Device Management.....	36
10.1	GetRfIPDevices.....	36
10.2	GetRfIPDevice	36
10.3	AddRfIPDevice.....	36
10.4	UpdateRfIPDevice.....	36
10.5	DeleteRfIPDevice	37
10.6	EnableRfIPDevice	37
10.7	ValidateRfIPDeviceConfig	37
10.8	SynchDeviceTimeWithHost	37
10.9	GetRfIPDeviceStatus	38
10.10	GetRfIPReaderStatus	38
10.11	StartRfIPDeviceReader	38
10.12	StopRfIPDeviceReader	38
10.13	GetControllerServiceStatus	39
10.14	Beep	39
10.15	DeletePendingData	39
10.16	RebootRfIPDevice.....	39
10.17	ShutdownRfIPDevice	39
10.18	StopRfIPDeviceService	40
10.19	StartRfIPDeviceService.....	40
10.20	RestartRfIPDeviceService	40
11	System Management	41
11.1	IsSystemRunning.....	41
11.2	SetSystemRunning.....	41
11.3	Ping.....	41
11.4	SynchronizeAllDeviceTimes.....	41
11.5	GetDevicePingTime	42
11.6	SystemStart	42
11.7	SystemStart	42
12	Tag Group Management	43
12.1	GetAllRFIDTagGroups	43
12.2	GetNumRFIDTagGroups	43
12.3	AddRFIDTagGroup.....	43
12.4	UpdateRFIDTagGroup	43
12.5	DeleteRFIDTagGroup	44
12.6	GetRFIDTagGroupByID	44

12.7	GetRFIDTagGroupByName	44
12.8	CleanUpDeletedTagGroups	44
13	RFID Tag Management.....	46
13.1	GetAllRFIDTags	46
13.2	GetNumRFIDTags.....	46
13.3	GetRFIDTagsInGroup.....	46
13.4	GetUnassignedRFIDTagsInGroup	47
13.5	GetRFIDTagGroupRelationships	47
13.6	GetNumberOfRFIDTagGroupRelationships.....	47
13.7	GetRFIDTag	48
13.8	GetRFIDTagByID	48
13.9	UndeleteRFIDTag	48
13.10	DeleteRFIDTag.....	48
13.11	AddRFIDTag	48
13.12	UpdateRFIDTag.....	49
13.13	CleanUpDeletedTags.....	49
14	World Location Management.....	50
14.1	GetWorldLocation	50
14.2	GetWorldLocations.....	50
14.3	DeleteWorldLocation	50
14.4	GetWorldLocationTypeDesc	50
14.5	GetWorldLocationsByTypeName.....	51
14.6	GetChildWorldLocations.....	51
14.7	GetLocationTypes	51
14.8	AddWorldLocation.....	51
14.9	UpdateWorldLocation	52
14.10	GetWorldHierarchyForLocation.....	52
14.11	GetWorldLocationPath	52
14.12	GetWorldLocationForVirtualLocation	53
15	Web Service Data Types	54
15.1	Asset	54
15.2	AssetAttribute.....	54
15.3	AssetLocationHistoryInfo	54
15.4	AssetLocationInfo	55
15.5	AssetLocationReportItem.....	55
15.6	AssetRestriction	56
15.7	AssetSearchDetail	56
15.8	AssetSearchTotal	56
15.9	AttribSearchCriterion	56
15.10	Category.....	56
15.11	CategoryAttribute	57
15.12	Department	57
15.13	HelpTip.....	57

15.14	LastKnownTagPosition	57
15.15	Location	57
15.16	LocationType	58
15.17	LogEntry.....	58
15.18	RFIDReader	58
15.19	RFIDReaderManufacturer	59
15.20	RFIDReaderModel.....	59
15.21	RFIDTag.....	59
15.22	RFIDTagAssetRelationship.....	59
15.23	RFIDTagGroup	60
15.24	RFIDTagGroupRelationship.....	60
15.25	RfIPDevice	60
15.26	RfIPDeviceLogEntry	60
15.27	WorldLocation	61
15.28	WorldLocationImage.....	61
15.29	WorldLocationRelationship	61
16	Enumerations	63
16.1	ComparisonType.....	63
16.2	DatabaseItemState	63
16.3	VirtualLocationType	63

1 Summary

The RfIP Web Service Application Programming Interface (API) is the mechanism through which 3rd party applications can interact with the RfIP Framework. The web service API provides services such as system, asset, RfIP device, and data management. The web service is accessible through a standard HTTP interface that is defined using the Web Service Description Language (WSDL). Any system or programming language that can format XML and post it to a web site can interact with the RfIP Web Service API.

2 Asset Management

The asset management methods allow 3rd party applications to perform tasks such as adding, locating, and deleting assets from the system. The methods also allow for assets' various attributes to be changed as necessary.

2.1 GetAsset

Summary:	Gets an asset.		
Parameters:	Name	Type	Comment
	aAssetID	Guid	The asset's unique system identifier
Returns:	Type	Comment	
	Asset	The specified asset or null if not found.	

2.2 GetNumAssets

Summary:	Gets the number of assets in the specified state.		
Parameters:	Name	Type	Comment
	aState	DatabaseItemState	Database state of the assets: Current, Deleted, All
Returns:	Type	Comment	
	int	The number of assets in the specified state.	

2.3 GetAssetsInCategory

Summary:	Gets assets that are members of a specified category.		
Parameters:	Name	Type	Comment
	aCategoryID	Guid	The category's unique system identifier
Returns:	Type	Comment	
	Asset[]	A list of assets in the specified category.	

2.4 GetAssetsInDepartment

Summary:	Gets all the assets that are associated with the specified department.		
Parameters:	Name	Type	Comment
	aDepartmentID	Guid	The unique system ID for the department in question.
Returns:	Type	Comment	
	Asset[]	A list of assets in the requested department.	

2.5 GetAssetByInventoryID

Summary:	Gets an asset by its inventory ID		
Parameters:	Name	Type	Comment
	aInventoryID	String	The asset's unique inventory identifier.
Returns:	Type	Comment	
	Asset	The specified asset or null if not found.	

2.6 DeleteAsset

Summary:	Deletes an asset.		
Parameters:	Name	Type	Comment
	aAssetID	Guid	The asset's unique system identifier.
	aFullDelete	Bool	Set to true to fully delete the asset from the system. Set to false to mark it deleted.
Returns:	Nothing		

2.7 AddAsset

Summary:	Adds an asset to the system.		
Parameters:	Name	Type	Comment
	aAssetID	Guid	The asset's unique system identifier.
	aRFIDTagID	Guid	The tag id to associate this asset with; empty Guid for null
	aCategoryID	Guid	The category id to associate this asset with; empty Guid for null
	aInventoryID	string	The unique inventory identifier for this asset
	aDepartmentID	Guid	The department id to associate this asset with; empty Guid for null
	aName	String	The name of the asset.
aDescription	String	A description of the asset.	
Returns:	Nothing		

2.8 UpdateAsset

Summary:	Updates an asset.		
Parameters:	Name	Type	Comment
	aAssetID	Guid	The asset's unique system identifier.
	aRFIDTagID	Guid	The tag id to associate this asset with; empty Guid for null
	aInventoryID	string	The unique inventory identifier for this

			asset
	aCategoryID	Guid	The category id to associate this asset with; empty Guid for null
	aDepartmentID	Guid	The department id to associate this asset with; empty Guid for null
	aName	String	The name of the asset.
	aDescription	String	A description of the asset.
Returns:	Nothing		

2.9 RemoveAssetFromDepartment

Summary:	Removes an asset from being associated with a department		
Parameters:	Name	Type	Comment
	aAssetID	Guid	The asset's unique system identifier.
Returns:	Nothing		

2.10 GetAssetForTag

Summary:	Gets the asset associated with the specified tag.		
Parameters:	Name	Type	Comment
	aTagID	Guid	The tag's unique system identifier.
Returns:	Type	Comment	
	Asset	The specified asset or null if not found.	

2.11 GetAssetLocation

Summary:	Gets the current location of an asset.		
Parameters:	Name	Type	Comment
	aAssetID	Guid	The unique system identifier for the asset.
	aTimeStamp	DateTime	The time in question.
Returns:	Type	Comment	
	AssetLocationHistoryInfo	Information about the location of the asset at the time specified.	

2.12 CleanUpDeletedAssets

Summary:	Permanently removes all assets that have been marked deleted from the system.		
Parameters:	None		
Returns:	Type	Comment	
	int	Number of assets fully deleted.	

2.13 GetAssetAttribute

Summary:	Gets an asset attribute.		
Parameters:	Name	Type	Comment
	aAssetAttribID	Guid	The unique system identifier for the attribute.
Returns:	Type	Comment	
	AssetAttribute	The requested asset attribute	

2.14 GetAttributesForAsset

Summary:	Gets the list of attributes for the specified asset.		
Parameters:	Name	Type	Comment
	aAssetID	Guid	The unique system identifier for the asset.
Returns:	Type	Comment	
	AssetAttribute[]	Array of AssetAttributes for the asset	

2.15 UpdateAssetAttribute

Summary:	Sets the value of an asset attribute.		
Parameters:	Name	Type	Comment
	AssetAttribID	Guid	The unique system identifier for the asset attribute.
	Value	String	The value to set it to (at most 128 characters)
Returns:	Type	Comment	
	bool	True on success; false on failure	

3 Category Management

The category management methods allow 3rd party applications to perform tasks such as adding, deleting, and updating categories in the system.

3.1 GetCategory

Summary:	Gets a category.		
Parameters:	Name	Type	Comment
	aCategoryID	Guid	The category's unique system identifier
Returns:	Type	Comment	
	Category	The specified category or null if not found.	

3.2 GetNumCategories

Summary:	Gets the number of categories in the system.		
Parameters:	None		
Returns:	Type	Comment	
	int	The number of categories in the system.	

3.3 GetNumberOfAssetsInCategory

Summary:	Gets the number of assets in a specified category.		
Parameters:	Name	Type	Comment
	aCategoryID	Guid	The unique system identifier for the category in question.
Returns:	Type	Comment	
	int	The number of categories in the specified category.	

3.4 DeleteCategory

Summary:	Deletes a category and all its sub-categories.		
Parameters:	Name	Type	Comment
	aCategoryID	Guid	The category's unique system identifier.
Returns:	Nothing		

3.5 AddCategory

Summary:	Adds a category to the system.		
Parameters:	Name	Type	Comment

	aCategoryID	Guid	The category's unique system identifier.
	aParentCategoryID	Guid	The new category's parent
	aName	String	The name of the category.
	aDescription	String	A description of the category; null for no description.
Returns:	Nothing		

3.6 UpdateCategory

Summary:	Updates a category.		
Parameters:	Name	Type	Comment
	aCategoryID	Guid	The category's unique system identifier.
	aName	String	The name of the category.
	aDescription	String	A description of the category; null for no description
Returns:	Nothing		

3.7 AddCategoryAttribute

Summary:	Adds a category attribute to the specified category. When assets are then placed in this category, the asset will inherit the attribute.		
Parameters:	Name	Type	Comment
	aCatAttribID	Guid	The unique system identifier for the category attribute.
	aCategoryID	Guid	The unique system identifier for the category to which the attribute is to be added.
	aAttribName	String	The name of the category attribute
Returns:	Nothing		
Exception:	Type	Comment	
	ApplicationException	Thrown when an the name of the attribute is already used in the category or in one of its ancestor categories.	

3.8 DeleteCategoryAttribute

Summary:	Deletes a category attribute from the system. Deleting a category attribute will remove the attribute from all categories and delete any values set on assets in those categories. Use with caution.		
Parameters:	Name	Type	Comment
	aCatAttribID	Guid	The unique system identifier for the category attribute.
Returns:	Nothing		

3.9 GetCategoryAttribute

Summary:	Gets the specified attribute.		
Parameters:	Name	Type	Comment
	CatAttribID	Guid	The unique system identifier for the category attribute in question.
	aInclInheritedAttribs	Bool	True if attributes from ancestor categories should be returned.
Returns:	Type	Comment	
	CategoryAttribute	The category attribute requested or null if not found.	

3.10 GetAttributesForCategory

Summary:	Gets the attributes associated with a category.		
Parameters:	Name	Type	Comment
	aCategoryID	Guid	The unique system identifier for the category in question.
	aInclInheritedAttribs	Bool	True if attributes from ancestor categories should be returned.
Returns:	Type	Comment	
	CategoryAttribute[]	Array of category attributes associated with the category in question.	

3.11 GetInheritedAttributesForCategory

Summary:	Get attributes for this category that are only inherited.		
Parameters:	Name	Type	Comment
	aCategoryID	Guid	The unique system identifier for the category in question.
Returns:	Type	Comment	
	CategoryAttribute[]	Array of inherited category attributes associated with the category in question.	

4 Department Management

The department management methods allow 3rd party applications to perform tasks such as adding, updating, and deleting departments from the system. The methods also allow for assets to be grouped by department.

4.1 GetDepartment

Summary:	Gets a department.		
Parameters:	Name	Type	Comment
	aDepartmentID	Guid	The department's unique system identifier
Returns:	Type	Comment	
	Department	The specified department or null if not found.	

4.2 GetDepartments

Summary:	Gets all departments.		
Parameters:	None		
Returns:	Type	Comment	
	Department[]	Array of all the departments in the system	

4.3 GetNumDepartments

Summary:	Gets the number of departments in the system.		
Parameters:	None		
Returns:	Type	Comment	
	int	The number of departments in the system.	

4.4 GetDepartmentByName

Summary:	Gets a department based on its name.		
Parameters:	Name	Type	Comment
	aDepartmentName	String	The department's name

4.5 DeleteDepartment

Summary:	Deletes a department.		
Parameters:	Name	Type	Comment

	aDepartmentID	Guid	The department's unique system identifier.
Returns:	Nothing		

4.6 AddDepartment

Summary:	Adds a department to the system.		
Parameters:	Name	Type	Comment
	aDepartmentID	Guid	The department's unique system identifier.
	aName	String	The name of the department.
Returns:	Nothing		

4.7 UpdateDepartment

Summary:	Updates a department to the system.		
Parameters:	Name	Type	Comment
	aDepartmentID	Guid	The department's unique system identifier.
	aName	String	The name of the department.
Returns:	Nothing		

4.8 DeleteDepartmentWorldLocationMapping

Summary:	Deletes a department to world location mapping.		
Parameters:	Name	Type	Comment
	aDepartmentID	Guid	The department's unique system identifier.
	aWorldLocationID	Guid	The world location's unique system identifier.
Returns:	Nothing		

4.9 AddDepartmentWorldLocationMapping

Summary:	Adds a department to world location mapping. If the world location is a child of an already existing location, the transaction is ignored.		
Parameters:	Name	Type	Comment
	aDepartmentID	Guid	The department's unique system identifier.
	aWorldLocationID	Guid	The world location's unique system identifier.

Returns:	Nothing
-----------------	---------

4.10 GetWorldLocationMappingsForDepartment

Summary:	Gets the world location mappings for a department.		
Parameters:	Name	Type	Comment
	aDepartmentID	Guid	The department's unique system identifier.
Returns:	Type	Comment	
	DepartmentWorldLocationMap[]	Array of all world locations mapped to this department.	

5 Location Image Management

The location image management methods allow 3rd party applications to retrieve and manage location images used by the system.

5.1 GetLocationImage

Summary:	Gets an image with the location highlighted.		
Parameters:	Name	Type	Comment
	aLocationID	Guid	The virtual location's unique system identifier
Returns:	Type	Comment	
	byte[]	Array of bytes that represents a PNG formatted image with the location highlighted	

5.2 GetLocationMappingsImage

Summary:	Gets an image with all the child locations highlighted for the specified world location.		
Parameters:	Name	Type	Comment
	aWorldLocationID	Guid	The world location's unique system identifier
Returns:	Type	Comment	
	byte[]	Array of bytes that represents a PNG formatted image with all the child locations highlighted for the specified world location.	

5.3 GetImageIDForLocation

Summary:	Gets the image identifier for the location in question.		
Parameters:	Name	Type	Comment
	aLocationID	Guid	The virtual location's unique system identifier
Returns:	Type	Comment	
	guid	The unique system identifier for the image for the location in question	

5.4 GetImageCoordinatesForLocation

Summary:	Gets a list of points that make up the bounds for the location on the image.		
Parameters:	Name	Type	Comment
	aImageID	Guid	The image's unique system identifier

	aLocationID	Guid	The virtual location's unique system identifier
Returns:	Type	Comment	
	Point[]	Array of points that make up the bounds for the location on the image	

6 Location History Management

The location history management methods allow 3rd party applications to retrieve and historical information about the history of an asset.

6.1 GetAssetHistoryCount

Summary:	Gets the number of asset history entries in the system.	
Parameters:	None	
Returns:	Type	Comment
	int	The number of asset history entries in the system.

6.2 GetRawTagHistoryAtLocation

Summary:	Gets raw history for a tag at a specified location for range of time. Note: The raw information is not always recorded at all virtual locations. This is a configurable switch.		
Parameters:	Name	Type	Comment
	aTagID	guid	The RFID tag's unique system identifier
	aLocationID	guid	The virtual location's unique system identifier
	aStartTime	DateTime	The starting timestamp
	aEndTime	DateTime	The ending timestamp
	aCullTagsByMaxDBI	bool	Set to true if history entries should be removed if their DBI/SSI is out of the virtual location's allowable range.
Returns:	Type	Comment	
	RuntimeTagInfo[]	Array of raw beacon information	

6.3 GetRawTagHistory

Summary:	Gets raw history for a tag for range of time. Note: The raw information is not always recorded at all virtual locations. This is a configurable switch.		
Parameters:	Name	Type	Comment
	aDeviceID	guid	The device to query: empty guid for all devices
	aTagID	guid	The RFID tag's unique system identifier

	aStartTime	DateTime	The starting timestamp
	aEndTime	DateTime	The ending timestamp
	aCullTagsByMaxDBI	bool	Set to true if history entries should be removed if their DBI/SSI is out of the virtual location's allowable range.
Returns:	Type	Comment	
	RuntimeTagInfo[]	Array of raw beacon information	

6.4 GetLocationsVisibleByTag

Summary:	Gets all the locations that currently see the tag. Note: The raw information is not always recorded at all virtual locations. This is a configurable switch.		
Parameters:	Name	Type	Comment
	aTagID	guid	The RFID tag's unique system identifier
Returns:	Type	Comment	
	guid[]	Array of unique system identifiers that represent virtual locations that are currently seeing the tag	

6.5 GetTagsVisibleAtLocation

Summary:	Gets all the locations that currently see the tag. Note: The raw information is not always recorded at all virtual locations. This is a configurable switch.		
Parameters:	Name	Type	Comment
	aLocationID	guid	The virtual location's unique system identifier
	aCullHighDBIValues	Bool	True to disregard all raw tag data that is out of the acceptable DBI/SSI range for this virtual location.
Returns:	Type	Comment	
	guid[]	Array of unique system identifiers that represent the tags that are currently visible at the specified virtual location.	

6.6 GetAssetHistory

Summary:	Gets the history for an asset for a specified time period.		
Parameters:	Name	Type	Comment
	aAssetID	guid	The asset's unique system identifier
	aStartDate	DateTime	The starting timestamp for the history to retrieve.

	aStopDate	DateTime	The ending timestamp for the history to retrieve.
Returns:	Type		Comment
	AssetLocationHistoryInfo[]	Array of history information items that represent the history (locations) of an asset for the specified time period.	

6.7 GetLastKnownTagPosition

Summary:	Gets the information about the last location a tag was seen. Note: The raw information is not always recorded at all virtual locations. This is a configurable switch.		
Parameters:	Name	Type	Comment
	aTagID	guid	The RFID tag's unique system identifier
Returns:	Type		Comment
	LastKnownTagPosition	Information about the last known position of an RFID tag.	

6.8 GetTagLocation

Summary:	Gets the location of a tag at the specified time. Note: The raw information is not always recorded at all virtual locations. This is a configurable switch.		
Parameters:	Name	Type	Comment
	aTagID	guid	The RFID tag's unique system identifier
	aTimeStamp	DateTime	The date and time of when to locate the tag.
Returns:	Type		Comment
	TagHistoryData	Information about the position of an RFID tag.	

6.9 GetLocationOfAssets

Summary:	Gets the location and status of assets in a category and child categories if requested.		
Parameters:	Name	Type	Comment
	aCategoryID	guid	A category's unique system identifier. This is the top level category if also including sub categories.
	aDepartmentID	guid	A department's unique system

			identifier; set to empty GUID for all departments.
	aTimeStamp	DateTime	A date and time of when to locate the assets in question.
	aIncludeChildCategories	bool	True if assets from child categories of the category represented by aCategoryID should be included in the search.
Returns:	Type		Comment
	AssetLocationReportItem[]		Array of report items, one for each asset contained in the search, which shows the location and status of each asset.

6.10 GetLastLocationOfMissingAssets

Summary:	Gets the location and status of any assets that are missing in a category and child categories if requested.		
Parameters:	Name	Type	Comment
	aCategoryID	guid	A category's unique system identifier. This is the top level category if also including sub categories.
	aDepartmentID	guid	A department's unique system identifier; set to empty GUID for all departments.
	aTimeStamp	DateTime	A date and time of when to locate the assets in question.
	aIncludeChildCategories	bool	True if assets from child categories of the category represented by aCategoryID should be included in the search.
Returns:	Type		Comment
	AssetLocationReportItem[]		Array of report items, one for each asset contained in the search, which shows the last known location and status of each asset.

6.11 GetInventoryAtLocation

Summary:	Gets the items that are currently located at the specified location		
Parameters:	Name	Type	Comment
	aLocationID	guid	A virtual location's unique system identifier.
Returns:	Type		Comment
	AssetLocationReportItem[]		Array of report items, one for each asset contained in the search, which shows the

LRNI

Lost Recovery Network, Inc.

		status of each asset.
--	--	-----------------------

7 Virtual Location Management

The virtual location management methods allow 3rd party applications to add, remove, and update virtual locations in the system.

7.1 GetLocation

Summary:	Gets a virtual location		
Parameters:	Name	Type	Comment
	aLocationID	guid	The virtual location's unique system identifier
Returns:	Type	Comment	
	Location	The virtual location in question or null if not found.	

7.2 DeleteLocation

Summary:	Deletes a virtual location		
Parameters:	Name	Type	Comment
	aLocationID	guid	The virtual location's unique system identifier
Returns:	Nothing		

7.3 GetLocations

Summary:	Gets virtual locations based on the system state specified		
Parameters:	Name	Type	Comment
	aState	DatabaseItemState	Virtual location system state filter: Current, Deleted, All
Returns:	Type	Comment	
	Location[]	The virtual locations in the specified state or null if none found.	

7.4 AddLocation

Summary:	Adds a virtual location to the system.		
Parameters:	Name	Type	Comment
	aLocationID	guid	The unique system identifier for this virtual location.
	aPath	string	A unique path for this location.

	aDeviceID	guid	The device to which this virtual location is associated.
	aName	string	The name of the virtual location.
	aMaxDBI	int	The maximum DBI/SSI allowed for beacons at this virtual location. -1 for no max.
	aPublishBeacons	bool	True if every beacon at this virtual location should be published
	aLocationType	VirtualLocationType	The type of virtual location.
	aEnabled	bool	True if the location is enabled, false otherwise.
Returns:	Type		Comment
	bool	True on success, false otherwise	

7.5 UpdateLocation

Summary:	Updates a virtual location in the system.			
Parameters:	Name		Type	Comment
	aLocationID	guid	The unique system identifier for this virtual location.	
	aPath	string	A unique path for this location.	
	aDeviceID	guid	The device to which this virtual location is associated.	
	aName	string	The name of the virtual location.	
	aMaxDBI	int	The maximum DBI/SSI allowed for beacons at this virtual location. -1 for no max.	
	aEnabled	bool	True if the location is enabled, false otherwise.	
	aPublishBeacons	bool	True if every beacon at this virtual location should be published	
aLocationType	VirtualLocationType	The type of virtual location.		
Returns:	Type		Comment	
	bool	True on success, false otherwise		

7.6 GetLocationByPath

Summary:	Gets a virtual location based on its path			
Parameters:	Name		Type	Comment

	aLocationPath	string	A unique path for this virtual location.
Returns:	Type		Comment
	Location	The virtual location that is referenced by aLocationPath or null if the location is not found in the system.	

7.7 GetUnboundLocations

Summary:	Gets a list of virtual locations that are not currently associated with a world location.		
Parameters:	None		
Returns:	Type		Comment
	Location[]	Array of virtual locations that are not currently associated with a world location.	

7.8 GetLocationsForDevice

Summary:	Gets a list of virtual locations associated with a device.		
Parameters:	Name	Type	Comment
	aDeviceID	Guid	The unique system identifier of the device in question.
Returns:	Type		Comment
	Location[]	Array virtual locations that are associated with the specified RfIP device.	

8 Log Management

The log management methods allow 3rd party applications to make entries or purge the log. There are two logs: the RfIP device log and the Monarch log. The RfIP device log should contain log entries related to device specific issues. The Monarch log should contain any other entries.

8.1 GetNumberOfRfIPDeviceLogEntries

Summary:	Gets the number of rfip device log entries based on the severity. This method does not use the aStartIndex or the aPageSize parameters there are only there for binding to the .NET ObjectDataSource component. If you wish to use this method, pass zero for both unused parameters.		
Parameters:	Name	Type	Comment
	aSeverityFilter	LogEntrySeverity	The types of log entries to count.
	aStartIndex	int	<i>Reserved</i>
	aPageSize	int	<i>Reserved</i>
Returns:	Type	Comment	
	int	The number of entries in the RfIP device log for the specified severity level.	

8.2 GetNumberOfMonarchLogEntries

Summary:	Gets the number of Monarch log entries based on the severity. This method does not use the aStartIndex or the aPageSize parameters there are only there for binding to the .NET ObjectDataSource component. If you wish to use this method, pass zero for both unused parameters.		
Parameters:	Name	Type	Comment
	aSeverityFilter	LogEntrySeverity	The types of log entries to count.
	aStartIndex	int	<i>Reserved</i>
	aPageSize	int	<i>Reserved</i>
Returns:	Type	Comment	
	int	The number of entries in the Monarch log for the specified severity level.	

8.3 LogRfIPDeviceMessage

Summary:	Logs a message for a specified RfIP device into the RfIP device log.
-----------------	--

Parameters:	Name	Type	Comment
	aRfIPDeviceID	guid	The unique system identifier for the RfIP device from which the message originated.
	aPostedDate	DateTime	Date the message was posted.
	aSeverity	string	The severity of the log entry: DEBUG, INFO, WARN, FATAL, ERROR
	aLogger	string	The name of the component from which the message originated.
	aMessage	string	The message to be logged.
	aException	string	Any exception information associated with the message.
Returns:	Nothing		

8.4 LogMonarchMessage

Summary:	Logs a message to the Monarch Log.		
Parameters:	Name	Type	Comment
	aPostedDate	DateTime	Date the message was posted.
	aSeverity	LogEntrySeverity	The severity of the log entry: DEBUG, INFO, WARN, FATAL, ERROR
	aLogger	string	The name of the component from which the message originated.
	aMessage	string	The message to be logged.
	aException	string	Any exception information associated with the message.
Returns:	Nothing		

8.5 PurgeMonarchLogEntries

Summary:	Deletes all log entries from the Monarch Log.		
Parameters:	None		
Returns:	Type	Comment	
	int	The number of entries deleted from the Monarch log.	

8.6 PurgeRfIPDeviceLogEntries

Summary:	Deletes all log entries from the RfIP device Log.	
Parameters:	None	
Returns:	Type	Comment
	int	The number of entries deleted from the RfIP device log.

8.7 GetRfIPDeviceLogEntries

Summary:	Gets the RfIP Device log entries based on the severity.		
Parameters:	Name	Type	Comment
	aSeverityFilter	LogEntrySeverity	The types of log entries to count.
Returns:	Type	Comment	
	RfIPDeviceLogEntry[]	Array of log entries from the RfIP Device log based on the severity filter.	

8.8 GetRfIPDeviceLogEntriesByPage

Summary:	Gets the RfIP Device log entries based on the severity. To access the fourth page with a page size of 10, aStartIndex should be set to 40 and aPageSize should be set to 10. aStartIndex is zero based.		
Parameters:	Name	Type	Comment
	aSeverityFilter	LogEntrySeverity	The types of log entries to count.
	aStartIndex	int	The zero based index of the page of entries to access.
	aPageSize	int	The number of entries to return per page.
Returns:	Type	Comment	
	RfIPDeviceLogEntry[]	Array of log entries from the RfIP Device log based on the severity filter. The maximum number of entries returned is give by aPageSize.	

8.9 GetMonarchLogEntries

Summary:	Gets the Monarch log entries based on the severity.		
Parameters:	Name	Type	Comment
	aSeverityFilter	LogEntrySeverity	The types of log entries to count.
Returns:	Type	Comment	
	LogEntry[]	Array of log entries from the Monarch log based on the severity filter.	

8.10 GetMonarchLogEntriesByPage

Summary:	Gets the RfIP Device log entries based on the severity. To access the fourth page with a page size of 10, aStartIndex should be set to 40 and aPageSize should be set to 10. aStartIndex is zero based.		
Parameters:	Name	Type	Comment
	aSeverityFilter	LogEntrySeverity	The types of log entries to count.
	aStartIndex	int	The zero based index of the page of entries to access.
	aPageSize	int	The number of entries to return per page.
Returns:	Type	Comment	
	LogEntry[]	Array of log entries from the Monarch log based on the severity filter. The maximum number of entries returned is give by aPageSize.	

9 RFID Reader Management

The RFID reader management methods allow 3rd party applications to control and manipulate RFID readers attached to the system.

9.1 GetRFIDReaders

Summary:	Gets a list of all of the standalone RFID Readers in the system.		
Parameters:	None		
Returns:	Type	Comment	
	RFIDReader[]	Array of standalone RFID readers in the system.	

9.2 GetRFIDReader

Summary:	Gets a standalone RFID reader.		
Parameters:	Name	Type	Comment
	aReaderID	guid	The unique system identifier for the standalone reader.
Returns:	Type	Comment	
	RFIDReader	The RFID reader specified by aReaderID or null if it is not found in the system.	

9.3 DeleteRFIDReader

Summary:	Deletes a standalone RFID reader.		
Parameters:	Name	Type	Comment
	aReaderID	guid	The unique system identifier for the standalone reader.
Returns:	Nothing		

9.4 AddRFIDReader

Summary:	Adds a standalone RFID reader to the system.		
Parameters:	Name	Type	Comment
	aReaderID	guid	The unique system identifier for the standalone RFID reader.
	aReaderTypeID	int	The unique system identifier for the type of standalone RFID reader.
	aName	string	The name of the standalone RFID reader.
	aEnabled	bool	True to enable the reader, False to

			disable it.
	aConfigParams	string	Configuration parameters for the reader.
Returns:	Type		Comment
	bool		True on success; false on failure

9.5 UpdateRFIDReader

Summary:	Updates a standalone RFID reader.		
Parameters:	Name		Type
			Comment
	aReaderID	guid	The unique system identifier for the standalone RFID reader.
	aReaderTypeID	int	The unique system identifier for the type of standalone RFID reader.
	aName	string	The name of the standalone RFID reader.
	aEnabled	bool	True to enable the reader, False to disable it.
	aConfigParams	string	Configuration parameters for the reader.
Returns:	Type		Comment
	bool		True on success; false on failure

9.6 GetRFIDReaderManufacturers

Summary:	Gets a list of all of the RFID reader manufacturers supported by the system.		
Parameters:	None		
Returns:	Type		Comment
	RFIDReaderManufacturer[]		Array of all of the RFID reader manufacturers supported by the system.

9.7 GetRFIDReaderModelsForManufacturer

Summary:	Gets a list of all of the RFID reader models for a particular manufacturer supported by the system.		
Parameters:	Name		Type
			Comment
	aManufacturerID	guid	The unique system identifier for the standalone RFID reader manufacturer.
Returns:	Type		Comment
	RFIDReaderModel[]		Array of all of the RFID reader models supported by the system for the specified manufacturer.

9.8 GetRFIDReaderManufacturerForModel

Summary:	Gets the manufacturer of a specified model of RFID reader.		
-----------------	--	--	--

Parameters:	Name	Type	Comment
	aRFIDReaderModelID	guid	The unique system identifier for the standalone RFID reader model.
Returns:	Type	Comment	
	RFIDReaderManufacturer	The manufacturer information for the model in question.	

9.9 GetRFIDReaderModel

Summary:	Gets standalone RFID reader model information.		
Parameters:	Name	Type	Comment
	aRFIDReaderModelID	guid	The unique system identifier for the standalone RFID reader model.
Returns:	Type	Comment	
	RFIDReaderModel	The RFID reader model information	

9.10 GetRFIDReaderStatus

Summary:	Gets the status of an standalone RFID reader.		
Parameters:	Name	Type	Comment
	aReaderID	guid	The unique system identifier for the standalone RFID reader.
Returns:	Type	Comment	
	ReaderStatus	The status of a standalone RFID reader.	

9.11 TestReaderConnectivity

Summary:	Determines the connectivity to the reader.		
Parameters:	Name	Type	Comment
	aReaderID	guid	The unique system identifier for the standalone RFID reader.
Returns:	Type	Comment	
	bool	True if there is connectivity to the reader; false otherwise	

9.12 ValidateReaderConfigurationParameters

Summary:	Validates that the configuration parameters are good for the specified reader model.		
Parameters:	Name	Type	Comment
	aReaderModelID	guid	The unique system identifier for the

			standalone RFID reader model.
	aConfigParams	string	The parameters to validate.
Returns:	Type		Comment
	bool		True if the parameters are correctly formatted for the model of the reader in question; false otherwise

9.13 StartRFIDReader

Summary:	Starts a standalone RFID reader.		
Parameters:	Name	Type	Comment
	aReaderID	guid	The unique system identifier for the standalone RFID reader.
	aStartTimeout	int	The amount of time to wait for the reader to start in milliseconds before abandoning the start.
Returns:	Type		Comment
	bool		True if the reader was started successfully; false otherwise

9.14 StopRFIDReader

Summary:	Stops a standalone RFID reader.		
Parameters:	Name	Type	Comment
	aReaderID	guid	The unique system identifier for the standalone RFID reader.
	aStartTimeout	int	The amount of time to wait for the reader to stop in milliseconds before abandoning the stop.
Returns:	Type		Comment
	bool		True if the reader was stopped successfully; false otherwise

10 RfIP Device Management

The RfIP device management methods allow 3rd party applications to start, stop, and get status of RfIP devices configured in the system.

10.1 GetRfIPDevices

Summary:	Gets all RfIP devices in the system.		
Parameters:			
Returns:	Type	Comment	
	RfIPDevice[]	Array of RfIP device info for all RfIP device in the system.	

10.2 GetRfIPDevice

Summary:	Gets a RfIP device's information.		
Parameters:	Name	Type	Comment
	aRfIPDeviceID	guid	The unique system identifier for the RfIP device.
Returns:	Type	Comment	
	RfIPDevice	Information about the RfIP device in question.	

10.3 AddRfIPDevice

Summary:	Adds a RfIP device to the system.		
Parameters:	Name	Type	Comment
	aRfIPDeviceID	guid	The unique system identifier for the RfIP device.
	aDeviceName	string	The name for the RfIP device.
	aIPAddress	string	The ip address assigned to this device, this can be null if unknown.
	aDeviceConfiguration	string	The xml configuration for this device.
Returns:	Nothing		

10.4 UpdateRfIPDevice

Summary:	Updates a RfIP device in the system.		
Parameters:	Name	Type	Comment
	aRfIPDeviceID	guid	The unique system identifier for the RfIP device to update.
	aDeviceName	string	The name for the RfIP device.

	aEnabled	bool	True to enable the device, false to disable the device.
	aIPAddress	string	The ip address assigned to this device, this can be null if unknown.
	aDeviceConfiguration	string	The xml configuration for this device.
Returns:	Nothing		

10.5 DeleteRfIPDevice

Summary:	Deletes a RfIP device from the system.		
Parameters:	Name	Type	Comment
	aRfIPDeviceID	guid	The unique system identifier for the RfIP device.
Returns:	Nothing		

10.6 EnableRfIPDevice

Summary:	Enables or disables a RfIP device.		
Parameters:	Name	Type	Comment
	aRfIPDeviceID	guid	The unique system identifier for the RfIP device.
	aState	bool	True = enable, False = disable
Returns:	Nothing		

10.7 ValidateRfIPDeviceConfig

Summary:	Validates the format of RfIP device configuration information.		
Parameters:	Name	Type	Comment
	aConfiguration	string	The xml configuration to validate.
Returns:	Type	Comment	
	bool	True if the configuration is valid; false otherwise	

10.8 SynchDeviceTimeWithHost

Summary:	Sends a message to a RfIP device to tell it to synchronize its time with the specified computer using the NNTP protocol.		
Parameters:	Name	Type	Comment
	aRfIPDeviceID	guid	The unique system identifier for the RfIP device.
	aHostName	string	Computer to synchronize with.
Returns:	Type	Comment	
	bool	True on success; false otherwise	

10.9 GetRfIPDeviceStatus

Summary:	Gets the overall status of a RfIP device.		
Parameters:	Name	Type	Comment
	aRfIPDeviceID	guid	The unique system identifier for the RfIP device.
Returns:	Type	Comment	
	bool	True if all device components are healthy; false otherwise	

10.10 GetRfIPReaderStatus

Summary:	Gets the status of a reader connected to a RfIP device.		
Parameters:	Name	Type	Comment
	aRfIPDeviceID	guid	The unique system identifier for the RfIP device.
Returns:	Type	Comment	
	bool	True if the reader is running; false otherwise	

10.11 StartRfIPDeviceReader

Summary:	Starts the RFID reader connected to a RfIP device.		
Parameters:	Name	Type	Comment
	aRfIPDeviceID	guid	The unique system identifier for the RfIP device.
Returns:	Type	Comment	
	bool	True if the reader is started successfully; false otherwise	

10.12 StopRfIPDeviceReader

Summary:	Stop the RFID reader connected to a RfIP device.		
Parameters:	Name	Type	Comment
	aRfIPDeviceID	guid	The unique system identifier for the RfIP device.
Returns:	Type	Comment	
	bool	True if the reader is stopped successfully; false otherwise	

10.13 GetControllerServiceStatus

Summary:	Gets the status of the RfIP Controller Service for a RfIP device.		
Parameters:	Name	Type	Comment
	aRfIPDeviceID	guid	The unique system identifier for the RfIP device.
Returns:	Type	Comment	
	bool	True if the service is running; false otherwise	

10.14 Beep

Summary:	Causes a RfIP device to beep for 1 minute.		
Parameters:	Name	Type	Comment
	aRfIPDeviceID	guid	The unique system identifier for the RfIP device.
Returns:	Type	Comment	
	bool	True if the beep was started successfully; false otherwise	

10.15 DeletePendingData

Summary:	Cleans up all cached data from a specified RfIP device.		
Parameters:	Name	Type	Comment
	aRfIPDeviceID	guid	The unique system identifier for the RfIP device.
Returns:	Type	Comment	
	bool	True if the pending data was deleted successfully; false otherwise	

10.16 RebootRfIPDevice

Summary:	Reboots a RfIP device.		
Parameters:	Name	Type	Comment
	aRfIPDeviceID	guid	The unique system identifier for the RfIP device.
Returns:	Type	Comment	
	bool	True if the reboot command was sent to the device successfully; false otherwise	

10.17 ShutdownRfIPDevice

Summary:	Shutsdown a RfIP device. If this method is called, the device will be no longer manageable over the network...use carefully.		
Parameters:	Name	Type	Comment
	aRfIPDeviceID	guid	The unique system identifier for the RfIP device.
Returns:	Type	Comment	
	bool	True if the shutdown command was sent to the device successfully; false otherwise	

10.18 StopRfIPDeviceService

Summary:	Stops the RfIP Controller Service for a RfIP device.		
Parameters:	Name	Type	Comment
	aRfIPDeviceID	guid	The unique system identifier for the RfIP device.
Returns:	Type	Comment	
	bool	True if service was stopped successfully; false otherwise	

10.19 StartRfIPDeviceService

Summary:	Starts the RfIP Controller Service for a RfIP device.		
Parameters:	Name	Type	Comment
	aRfIPDeviceID	guid	The unique system identifier for the RfIP device.
Returns:	Type	Comment	
	bool	True if service was started successfully; false otherwise	

10.20 RestartRfIPDeviceService

Summary:	Stops and then restarts the RfIP Controller Service for a RfIP device.		
Parameters:	Name	Type	Comment
	aRfIPDeviceID	guid	The unique system identifier for the RfIP device.
Returns:	Type	Comment	
	bool	True if service was stopped and restarted successfully; false otherwise	

11 System Management

11.1 IsSystemRunning

Summary:	Gets a flag that shows if the system is turned on or off. This flag only changes if set programmatically. If a part of the system fails, this value will not be automatically changed to false.	
Parameters:	None	
Returns:	Type	Comment
	bool	True if the system is running; false otherwise

11.2 SetSystemRunning

Summary:	Sets a flag that shows if the system is turned on or off. This flag only changes if set programmatically. If a part of the system fails, this value will not be automatically changed to false.		
Parameters:	Name	Type	Comment
	aState	bool	True to set the running flag to true; false to set the running flag to false
Returns:	Nothing		

11.3 Ping

Summary:	Pings an IP address.		
Parameters:	Name	Type	Comment
	aIPAddress	string	The IP address of the network node to ping.
Out Parameters:	Name	Type	Comment
	aMsg	string	The message that represents the state of the ping.
Returns:	Type	Comment	
	bool	True if the ping was successful; false otherwise	

11.4 SynchronizeAllDeviceTimes

Summary:	Synchronizes all device times with the server.		
Parameters:	None		
Returns:	Type	Comment	
	bool	True if all the devices' times were synchronized	

	successfully; false otherwise
--	-------------------------------

11.5 GetDevicePingTime

Summary:	Gets the average round trip ping time for the specified IP address.		
Parameters:	Name	Type	Comment
	aIPAddress	string	The IP address of the network node to ping.
	aNumPings	int	The number of times to ping the device.
Returns:	Type	Comment	
	int	The average round trip ping time.	

11.6 SystemStart

Summary:	Starts the system and all RfIP devices.		
Parameters:	None		
Returns:	Type	Comment	
	bool	True on success; false otherwise	

11.7 SystemStart

Summary:	Stops the system and all RfIP devices.		
Parameters:	None		
Returns:	Nothing		

12 Tag Group Management

All tags are separated into groups based on a tag namespace or manufacturer. These groupings are managed via the tag group management methods in the RfIP web service.

12.1 GetAllRFIDTagGroups

Summary:	Gets a list of the RFID Tag Groups currently in the system based on a state filter.		
Parameters:	Name	Type	Comment
	aState	DatabasItemState	The tag group state filter: Deleted, All, Current
Returns:	Type	Comment	
	RFIDTagGroup[]	Array of RFID tag groups in the system filtered by the state requested.	

12.2 GetNumRFIDTagGroups

Summary:	Gets the number RFID Tag Groups currently in the system based on a state filter.		
Parameters:	Name	Type	Comment
	aState	DatabasItemState	The tag group state filter: Deleted, All, Current
Returns:	Type	Comment	
	int	Number of RFID tag groups in the system filtered by the state requested.	

12.3 AddRFIDTagGroup

Summary:	Adds an RFID tag group to the system.		
Parameters:	Name	Type	Comment
	aTagGroupID	guid	The unique system identifier for this tag group.
	aTagGroupName	string	The name for this tag group.
Returns:	Nothing		

12.4 UpdateRFIDTagGroup

Summary:	Updates an RFID tag group already in the system.
-----------------	--

Parameters:	Name	Type	Comment
	aTagGroupID	guid	The unique system identifier for this tag group.
	aTagGroupName	string	The name for this tag group.
Returns:	Nothing		

12.5 DeleteRFIDTagGroup

Summary:	Deletes an RFID tag group from the system. The tag group can either be fully deleted or marked as deleted.		
Parameters:	Name	Type	Comment
	aTagGroupID	guid	The unique system identifier for this tag group.
	aFullDelete	bool	True to completely remove the tag group information from the database; False to mark the tag group as deleted in the database.
Returns:	Nothing		

12.6 GetRFIDTagGroupByID

Summary:	Gets an RFID tag group based on its unique system identifier.		
Parameters:	Name	Type	Comment
	aGroupID	guid	The unique system identifier for this tag group.
Returns:	Type	Comment	
	RFIDTagGroup	The RFID tag group in question or null if it is not found.	

12.7 GetRFIDTagGroupByName

Summary:	Gets an RFID tag group based on its unique name.		
Parameters:	Name	Type	Comment
	aGroupName	string	The unique name for this tag group.
Returns:	Type	Comment	
	RFIDTagGroup	The RFID tag group in question or null if it is not found.	

12.8 CleanUpDeletedTagGroups

Summary:	Permanently removes all RFID tag groups that are marked as deleted from the system.		
Parameters:	None		
Returns:	Type	Comment	

	int	The number of RFID Tag group permanently deleted from the system.
--	-----	---

13 RFID Tag Management

These methods allow 3rd party applications to add, delete, and update RFID tag information in the system.

13.1 GetAllRFIDTags

Summary:	Gets a list of the RFID tags currently in the system based on a state filter.		
Parameters:	Name	Type	Comment
	aState	DatabasItemState	The tag state filter: Deleted, All, Current
Returns:	Type	Comment	
	RFIDTag[]	Array of RFID tags in the system filtered by the state requested.	

13.2 GetNumRFIDTags

Summary:	Gets the number RFID tags currently in the system based on a state filter.		
Parameters:	Name	Type	Comment
	aState	DatabasItemState	The tag state filter: Deleted, All, Current
Returns:	Type	Comment	
	int	Number of RFID tags in the system filtered by the state requested.	

13.3 GetRFIDTagsInGroup

Summary:	Gets a list of the RFID tags currently in the system that belong to a certain RFID tag group and based on a state filter.		
Parameters:	Name	Type	Comment
	aGroupID	guid	The unique system identifier of the RFID tag group.
	aState	DatabasItemState	The tag state filter: Deleted, All, Current
Returns:	Type	Comment	
	RFIDTag[]	Array of RFID tags in the system filtered by the state requested that belong to the RFID tag group requested.	

13.4 GetUnassignedRFIDTagsInGroup

Summary:	Gets a list of the RFID tags currently in the system that are in a certain RFID tag group and are not assigned to an asset.		
Parameters:	Name	Type	Comment
	aGroupID	guid	The unique system identifier of the RFID tag group.
Returns:	Type	Comment	
	RFIDTag[]	Array of RFID tags in the system that belong to the RFID tag group requested and are not associated with an asset.	

13.5 GetRFIDTagGroupRelationships

Summary:	Gets a list of all the RFID Tags based on specified criteria.		
Parameters:	Name	Type	Comment
	aRangeStart	long	The starting tag ID – use 0 to include all tags.
	aTagGroupID	guid	The unique system identifier of the RFID tag group.
	aState	DatabasItemState	The tag state filter: Deleted, All, Current
Returns:	Type	Comment	
	RFIDTagGroupRelationship[]	Array of RFID tags in the that meet the specified criteria.	

13.6 GetNumberOfRFIDTagGroupRelationships

Summary:	Gets a list of all the RFID Tags based on specified criteria.		
	Not to be used by 3rd party applications. LRNI Internal Use only.		
Parameters:	Name	Type	Comment
	aStartTagID	long	The starting tag ID – use 0 to include all tags.
	aGroupIDFilter	guid	The unique system identifier of the RFID tag group.
	aState	DatabasItemState	The tag state filter: Deleted, All, Current
	aStartIndex	int	<i>Reserved</i> – set to 0
	aPageSize	int	<i>Reserved</i> – set to 0
Returns:	Type	Comment	
	int	The number of RFID Tags based on specified criteria.	

13.7 GetRFIDTag

Summary:	Gets an RFID tag.		
Parameters:	Name	Type	Comment
	aTagID	long	The tag identification number (not the system identifier)
	aTagGroupName	string	The name of the tag group to which this tag belongs.
Returns:	Type	Comment	
	RFIDTag	An RFID tag or null if not found.	

13.8 GetRFIDTagByID

Summary:	Gets an RFID tag.		
Parameters:	Name	Type	Comment
	aTagID	guid	The unique system identifier for the tag.
Returns:	Type	Comment	
	RFIDTag	An RFID tag or null if not found.	

13.9 UndeleteRFIDTag

Summary:	Undeletes an RFID tag that has previously been marked deleted.		
Parameters:	Name	Type	Comment
	aTagID	guid	The unique system identifier for the tag.
Returns:	Type	Comment	
	bool	True on success; false on error	

13.10 DeleteRFIDTag

Summary:	Deletes an RFID tag from the system completely or marks it as deleted.		
Parameters:	Name	Type	Comment
	aTagID	guid	The unique system identifier for the tag.
	aFullDelete	bool	If true, the tag is completely deleted from the database; if false, the tag is only marked as deleted.
Returns:	Nothing		

13.11 AddRFIDTag

Summary:	Adds an RFID tag to the system.
-----------------	---------------------------------

Parameters:	Name	Type	Comment
	aTagDBID	guid	The unique system identifier for this tag.
	aTagID	long	The identifier for the RFID tag (set by the manufacturer)
	aGroupID	guid	The unique system identifier for the tag group to associate the tag with.
	aEnabled	bool	True to enable the tag; false otherwise
aBatteryDate	DateTime	The date the battery was installed or max date for passive tags.	
Returns:	Nothing		

13.12 UpdateRFIDTag

Summary:	Updates an RFID tag already in the system.		
Parameters:	Name	Type	Comment
	aID	guid	The unique system identifier for this tag.
	aTagID	long	The identifier for the RFID tag (set by the manufacturer)
	aGroupID	guid	The unique system identifier for the tag group to associate the tag with.
	aEnabled	bool	True to enable the tag; false otherwise
aBatteryDate	DateTime	The date the battery was installed or max date for passive tags.	
Returns:	Type	Comment	
	RFIDTag	The updated RFID tag.	

13.13 CleanUpDeletedTags

Summary:	Permanently removes all RFID tags that are marked as deleted from the system.		
Parameters:	None		
Returns:	Type	Comment	
	int	The number of RFID tags permanently removed from the system.	

14 World Location Management

The world location management methods allow 3rd party applications to manage the layout of the world. World locations differ from virtual locations in that a virtual location maps to an antenna while a world location maps to a zone, room, building, etc.

14.1 GetWorldLocation

Summary:	Gets a world location.		
Parameters:	Name	Type	Comment
	aWorldLocationID	guid	The unique system identifier for the world location.
Returns:	Type	Comment	
	WorldLocation	The world location requested or null if not found.	

14.2 GetWorldLocations

Summary:	Gets a list of all the world locations in the system.		
Parameters:	None		
Returns:	Type	Comment	
	WorldLocation[]	Array of all of the world locations in the system.	

14.3 DeleteWorldLocation

Summary:	Deletes a world location.		
	This will also delete all of the child world locations of the world location.		
Parameters:	Name	Type	Comment
	aWorldLocationID	guid	The unique system identifier for the world location.
Returns:	Nothing		

14.4 GetWorldLocationTypeDesc

Summary:	Gets a world location.		
Parameters:	Name	Type	Comment
	aLocationTypeID	int	The unique system identifier for the world location type.

Returns:	Type	Comment
	string	The world location type description or null if not found.

14.5 GetWorldLocationsByTypeName

Summary:	Gets a list of world locations that are of a specific type.		
Parameters:	Name	Type	Comment
	aLocationTypeName	string	The name of the world location type.
Returns:	Type	Comment	
	WorldLocation[]	Array of world locations of the requested type.	

14.6 GetChildWorldLocations

Summary:	Gets a list of world locations that are a direct child of the specified world location.		
Parameters:	Name	Type	Comment
	aWorldLocationID	guid	The unique system identifier for the world location.
Returns:	Type	Comment	
	WorldLocation[]	Array of world locations that are direct children of the world location given by aWorldLocationID.	

14.7 GetLocationTypes

Summary:	Gets a list of all of the different location types in the system.		
Parameters:	None		
Returns:	Type	Comment	
	LocationType[]	Array of location types in the system.	

14.8 AddWorldLocation

Summary:	Adds a world location to the system.		
Parameters:	Name	Type	Comment
	aWorldLocationID	guid	The unique system identifier for the world location.
	aLocationID	guid	The unique system identifier for the virtual location to associate with this world location. Set to an empty GUID to disassociate from any virtual location.

	aLocationType	int	The unique system identifier for the world location type.
	aParentWorldLocation	guid	The unique system identifier for the parent world location.
	aName	string	The name of the world location.
Returns:	Type		Comment
	bool	True on success; false on failure	

14.9 UpdateWorldLocation

Summary:	Updates a world location in the system.		
Parameters:	Name		Type
			Comment
	aWorldLocationID	guid	The unique system identifier for the world location.
	aLocationID	guid	The unique system identifier for the virtual location to associate with this world location. Set to an empty GUID to disassociate from any virtual location.
	aName	string	The name of the world location.
Returns:	Type		Comment
	bool	True on success; false on failure	

14.10 GetWorldHierarchyForLocation

Summary:	Gets the hierarchy of the specified world location.		
Parameters:	Name		Type
			Comment
	aWorldLocationID	guid	The unique system identifier for the world location.
Returns:	Type		Comment
	WorldLocation[]	Array of world locations that represent the ancestors in order of the specified world location.	

14.11 GetWorldLocationPath

Summary:	Gets the text description of the hierarchy of the specified world location.		
Parameters:	Name		Type
			Comment
	aWorldLocationID	guid	The unique system identifier for the world location.
Returns:	Type		Comment
	string	A text description of the hierarchy of the specified world	

	location. Example: World – Chicago – Building A – Floor 1 – Room 106
--	---

14.12 GetWorldLocationForVirtualLocation

Summary:	Gets a world location that is associated with a specified virtual location.		
Parameters:	Name	Type	Comment
	aVirtualLocationID	guid	The unique system identifier for the virtual location.
Returns:	Type	Comment	
	WorldLocation	The world location that is associated with the specified virtual location or null if it is not associated with any world location.	

15 Web Service Data Types

15.1 Asset

Field	Type	Description
AssetID	guid	The unique system identifier for this asset.
CategoryID	guid	The unique system identifier for the category with which this asset is associated. An empty GUID represents an uncategorized asset.
DepartmentID	guid	The unique system identifier for the department with which this asset is associated. An empty GUID represents an asset that is not associated with a department.
RFIDTagID	guid	The unique system identifier for the RFID tag with which this asset is associated. An empty GUID represents an asset that is not associated with an RFID tag.
InventoryID	string	The unique system identifier for the inventory control number used to identify this asset. This is sometimes a barcode or a key into a 3 rd party database.
ShortDesc	string	The name of the asset.
LongDesc	string	A description of the asset.
DeletedDate	DateTime	If the asset has been marked deleted, this will be the timestamp of when it was marked deleted.
Deleted	bool	True if the asset has been marked deleted; false otherwise.

15.2 AssetAttribute

Field	Type	Description
AssetAttributeID	guid	The unique system identifier for this asset attribute.
AssetID	guid	The unique system identifier for the asset with which this attribute is associated.
CatAttribID	guid	The unique system identifier for the category attribute with which this asset attribute is associated.
AttribName	string	The name of the asset attribute.
Value	string	The value of the asset attribute or null if there is no value set.

15.3 AssetLocationHistoryInfo

Field	Type	Description
AssetID	guid	The unique system identifier of the Asset to which this history item belongs.
Visible	bool	A flag that signifies if this history item denotes that the asset was visible at this location or not visible at this location. <i>Not Used</i>
LocationID	guid	The unique system identifier of the virtual location where this

		asset is located according to this history item.
SecondaryLocationID	guid	The unique system identifier of the secondary virtual location where this asset is located according to this history item. This is the virtual location that the asset is closest to, but not in.
StartTime	DateTime	The timestamp that marks when the asset entered this virtual location.
EndTime	DateTime	The timestamp that marks when the asset exited this virtual location. If the asset is still at this location, the end time will be set to 12/31/9999 23:59:59.9999999.

15.4 AssetLocationInfo

Field	Type	Description
AssetID	guid	The unique system identifier of the Asset to which this history item belongs.
LocationID	guid	The unique system identifier of the virtual location where this asset is located according to this history item.
SecondaryLocationID	guid	The unique system identifier of the secondary virtual location where this asset is located according to this history item. This is the virtual location that the asset is closest to, but not in.
TimeStamp	DateTime	The timestamp that marks when the asset entered this virtual location.
SignalStrength	int	The DBI/SSI at which the asset's tag is being read at the virtual location.

15.5 AssetLocationReportItem

Field	Type	Description
AssetID	guid	The unique system identifier of the Asset to which this report item belongs.
InventoryID	string	The unique asset control number used to identify the asset.
AssetName	string	The name given to the asset. This is usually related to the type of asset or a very short description of the asset.
CategoryID	guid	The unique system identifier of the category to which this asset belongs.
CategoryName	guid	The name of the category to which this asset belongs.
LocationTypeDesc	string	A short description of the type of location where the asset resides (zone, room, section, floor, building, campus)
LocationID	guid	The unique system identifier of the virtual location where this asset is located according to this history item.
LocationName	guid	The name of the virtual location where the asset currently resides.
TimeStamp	DateTime	The time when the asset first entered this virtual location.
MissingStatus	bool	A flag that signifies if the asset is missing. If true, the asset is missing and all of the other location fields will provide the

		last virtual location that the asset was known to be.
--	--	---

15.6 AssetRestriction

Field	Type	Description
AssetID	guid	The unique system identifier of the Asset to which this restriction item belongs.
RestrictionID	guid	The unique system identifier of the restriction.
WorldLocationID	guid	The unique system identifier of the world location to which this restriction applies.
Deny	bool	If true, the asset is denied access to the world location; otherwise the asset only allowed to be in this world location.

15.7 AssetSearchDetail

Field	Type	Description
AssetID	guid	The unique system identifier of the Asset to which this search detail item belongs.
SearchTime	DateTime	The timestamp of when the search took place.
UserID	string	The identifier of the user who performed the search.

15.8 AssetSearchTotal

Field	Type	Description
AssetID	guid	The unique system identifier of the Asset to which this search total item belongs.
LastSearch	DateTime	The timestamp of when the last search took place.
SearchCount	int	The total number of times this item has been searched for.

15.9 AttribSearchCriterion

Field	Type	Description
CatAttribID	guid	The unique system identifier of the category attribute.
ComparisonType	ComparisonType	The type of comparison to perform.
Value	string	The value to compare against.

15.10 Category

Field	Type	Description
CategoryID	guid	The unique system identifier of the category.

ParentCategoryID	guid	The unique system identifier of this category's parent category. If the value is a empty GUID then the category has no parent.
ShortDesc	guid	The name of the category.
LongDesc	guid	A description of the category.

15.11 CategoryAttribute

Field	Type	Description
CategoryID	guid	The unique system identifier of the category.
CatAttribID	guid	The unique system identifier of the category attribute.
AttribName	string	The name of the category attribute.

15.12 Department

Field	Type	Description
DepartmentID	guid	The unique system identifier of the department.
DeptName	string	The name of the department.

15.13 HelpTip

Field	Type	Description
TipID	guid	The unique system identifier of the help tip.
IsAdminTip	bool	True if this tip applies to administrative users.
TipText	string	The help tip's message.
TipImage	byte[]	An image that is associated with this help tip.
TipLink	string	A reference to more detail about this help tip subject.

15.14 LastKnownTagPosition

Field	Type	Description
LocationID	guid	The unique system identifier of the virtual location for the RFID tag in question.
TimeStamp	DateTime	The last date and time that the RFID tag in question was seen.

15.15 Location

Field	Type	Description
LocationID	guid	The unique system identifier of the virtual location.

LocationType	VirtualLocationType	The type of virtual location.
PublishBeacons	bool	True if all tag beacons at this location should be publish back to the server; false otherwise.
Enabled	bool	True if this location is enabled; false otherwise.
MaxDBI	Int	The maximum DBI/SSI that is allowed to be reported from this virtual location.
PathInfo	string	The unique identifying path that can be used to lookup this virtual location.
ShortDesc	string	The name of this virtual location.
LongDesc	string	A description of this virtual location.
DeletedDate	DateTime	The date and time that this virtual location was marked as deleted.
Deleted	bool	True if this virtual location has been marked as deleted from the system; false otherwise.
DeviceID	guid	The unique system identifier of the device to which this virtual location is attached.

15.16 LocationType

The LocationType object is related to world locations and not to virtual locations.

Field	Type	Description
LocationTypeID	guid	The unique system identifier of this location type.
ShortDesc	string	A short description of the location type (world, campus, building, floor, section, room, zone, etc.)

15.17 LogEntry

Field	Type	Description
ID	guid	The unique system identifier of this log entry.
Severity	LogEntrySeverity	The severity of the log entry. (DEBUG, WARN, ERROR, FATAL, INFO)
Message	string	The log message.
Exception	string	The exception information associated with the log message.
Logger	string	The source of the message.
DatePosted	DateTime	The date and time of when the message was posted.

15.18 RFIDReader

Field	Type	Description
ID	guid	The unique system identifier of this standalone RFID reader.
ReaderTypeID	guid	The unique system identifier of this standalone RFID reader's type.

Name	string	The name of the standalone RFID reader.
ConfigParams	string	The configuration parameters used to configure and control the standalone RFID reader.
Enabled	bool	True if the reader is enabled; false otherwise.

15.19 RFIDReaderManufacturer

Field	Type	Description
ID	guid	The unique system identifier of this RFID reader manufacturer.
Name	string	The name of the RFID reader manufacturer.

15.20 RFIDReaderModel

Field	Type	Description
ID	guid	The unique system identifier of this RFID reader model.
ManufacturerID	guid	The unique system identifier of the manufacturer of this RFID reader model.
Name	string	The name of the RFID reader manufacturer.
DriverAssemblyName	string	The name of the .NET Assembly that contains the Type required to interface with this model of RFID reader.
DriverTypeName	string	The name of the .NET Type that is required to interface with this model of RFID reader.

15.21 RFIDTag

Field	Type	Description
ID	guid	The unique system identifier of this RFID Tag.
BatteryDate	DateTime	The date and time the battery was last replaced.
Enabled	bool	True if this location is enabled; false otherwise.
TagID	long	The unique identifier for this tag as set by the tag's manufacturer.
GroupID	guid	The unique system identifier of the RFID tag group to which this tag belongs.
DeletedDate	DateTime	The date and time that this RFID Tag was marked as deleted.
Deleted	bool	True if this RFID Tag has been marked as deleted from the system; false otherwise.

15.22 RFIDTagAssetRelationship

Field	Type	Description
AssetID	guid	The unique system identifier of the Asset.
AssetShortDesc	string	The name of the asset.

RFIDTagID	long	The unique identifier for this tag as set by the tag's manufacturer.
RFIDTagGroupID	string	The name of the RFID Tag group to which this tag belongs.

15.23 RFIDTagGroup

Field	Type	Description
ID	guid	The unique system identifier of this RFID Tag Group.
GroupID	guid	The unique manufacturer identifier of the RFID tag group.
DeletedDate	DateTime	The date and time that this RFID Tag Group was marked as deleted.
Deleted	bool	True if this RFID Tag Group has been marked as deleted from the system; false otherwise.

15.24 RFIDTagGroupRelationship

Field	Type	Description
TagDBID	guid	The unique system identifier of the RFID tag.
TagID	long	The unique identifier for this tag as set by the tag's manufacturer.
GroupDBID	guid	The unique system identifier of the RFID tag group.
GroupID	string	The unique manufacturer identifier given to the RFID tag group.
TagEnabled	bool	True if the RFID tag is enabled; false otherwise.
TagDeleted	bool	True if the RFID tag is marked deleted; false otherwise.

15.25 RfIPDevice

Field	Type	Description
DeviceID	guid	The unique system identifier of the RfIP device.
Enabled	bool	True if the RfIP device is enabled; false otherwise.
DeletedDate	DateTime	The date and time that this RfIP device was marked as deleted.
Deleted	bool	True if this RfIP device has been marked as deleted from the system; false otherwise.
DeviceName	string	The name of the RfIP device.
ConfigData	string	Configuration information for the RfIP device.
IPAddress	string	The IP address of the RfIP device.

15.26 RfIPDeviceLogEntry

Field	Type	Description
-------	------	-------------

ID	guid	The unique system identifier of this log entry.
Severity	LogEntrySeverity	The severity of the log entry. (DEBUG, WARN, ERROR, FATAL, INFO)
Message	string	The log message.
Exception	string	The exception information associated with the log message.
Logger	string	The source of the message.
DatePosted	DateTime	The date and time of when the message was posted.
RfIPDeviceID	guid	The unique system identifier of the RfIP device associated with this message.

15.27 WorldLocation

Field	Type	Description
WorldLocationID	guid	The unique system identifier of the world location.
LocationID	guid	The unique system identifier of the virtual location with which this world location is associated. An empty GUID signifies that there is no associated virtual location.
LocationTypeID	int	The unique system identifier of the location type.
ParentWorldLoationID	guid	The unique system identifier of the parent world location with which this world location is a child. An empty GUID signifies that there is no parent to the world location.
ShortDesc	string	The name of the world location.

15.28 WorldLocationImage

Field	Type	Description
ImageID	guid	The unique system identifier of the image.
ImagePath	string	The file system path to the image on disk.

15.29 WorldLocationRelationship

Field	Type	Description
WorldLocationID	guid	The unique system identifier of the world location.
ParentWorldLoationID	guid	The unique system identifier of the parent world location with which this world location is a child. An empty GUID signifies that there is no parent to the world location.
LocationTypeDesc	string	The description of the world location type.
VirtualLocation	string	The unique path to the virtual location with which this world location is associated.
VirtualLocationShortDesc	string	The name of the virtual location with which this world location is associated.
WorldLocationShortDesc	string	The name of the world location.

LRNI

Lost Recovery Network, Inc.

16 Enumerations

16.1 ComparisonType

Value	Description
Equal	Check if values are equal.
NotEqual	Check if values are not equal.
LessThan	Check if value A is less than value B.
LessThanOrEqual	Check if value A is less than or equal to value B.
GreaterThan	Check if value A is greater than value B.
GreaterThanOrEqual	Check if value A is greater than or equal to value B.

16.2 DatabaseItemState

Value	Description
Current	Selects items that are not marked as deleted in the system.
Deleted	Selects items that are marked as deleted in the system.
All	Selects all items in the database no matter what their state.

16.3 VirtualLocationType

Value	Description
RfIPAntenna	This virtual location represents an antenna connected to a RfIP device on the cycle side.
RfIPDedicatedAntenna	This virtual location represents an antenna connected to a RfIP device on the dedicated side.
ReaderAntenna	This virtual location represents an antenna connected to a standalone reader.
Navigation	This virtual location represents an navigation item used by the software.
Unknown	This virtual location has an unknown type.