

API Request Message

The API request messages are limited to a maximum of 64K in size by the .NET XML reader in order to insure that the request messages will remain below the .NET 85,000 byte limit and not end up being stored in the LOH.

API Request Message

```
<ReqMsg>
  <UserName />           - DGP system account name
  <ReqID />               - unique ID created by the client app for each request message, and echoed back in the response
  <ReqToken />            - HMAC hash of the Time value using the account password as the secret key
  <Time />                - UTC Unix time of the request for the TTL check and the HMAC hash authentication to the server
  <MList>                 - a collection of one or more API methods to be called
    <Meth>
      <MName />           - the name of the API method being called
      <PList>             - a collection of zero or more input parameters for the API method
        <Prm>             - name/value pairs for each input parameter
          <Name />         - the name of the input parameter for the API method
          <Val><![CDATA[ ... ]]></Val> - each input parameter value is encapsulated within a CDATA block
        </Prm>
      </PList>
    </Meth>
  </MList>
</ReqMsg>
```

API Response Message

The API response messages are limited to a maximum of 64K in size by the .NET XML reader in order to insure that the response messages will remain below the .NET 85,000 byte limit and not end up being stored in the LOH. Limiting the size of the response messages depends on the server-side pagination of tabular data.

API Response Message

```
<RespMsg>
  <UserName />           - DGP system account name
  <ReqID />               - unique ID created by the client app for each request message, and echoed back in the response
  <Time />                - UTC Unix time of the response
  <Auth />                - state of the request message authentication (OK, NoMatch, Expired, Disabled, Error, Exception)
  <Info />                - optional information regarding Auth states other than OK
  <SvrMS />               - the time spent on the server executing all of the API method calls in the request message batch
  <MethCount />           - the number of methods called in the request message batch
  <RList>
    <Result>
      <RName />           - the name of the method result, used by the client to match results to method calls
      <RCode />           - code indicating the state of the method result (OK, Empty, Error, Exception)
      <DType />           - the data type of the result value (Int, Num, Text, DateTime, XML, JSON, DataTable)
      <RVal><![CDATA[ ... ]]></Val> - each return value is encapsulated within a CDATA block
    </Result>
  </RList>
</RespMsg>
```

Schema and Table Names

Schema Name	Table Name	Description
DGPDrive	Favorites	REPLICA – files flagged by the user as favorites for easy navigation
DGPDrive	Files	HYBFILE – file info metadata and the file path to the file in storage
DGPDrive	FileTags	REPLICA – tags linked to a specific file metadata record
DGPDrive	Folders	REPLICA – folder (category tree) used to group file metadata records
DGPDrive	Tags	REPLICA – tags (category cloud) used to describe file metadata records
SysInfo	APIMethods	REPLICA – the list of all API methods in a system used for authorization
SysInfo	APIRoles	REPLICA – groups of methods used in RBAC
SysInfo	APIUsers	REPLICA – user accounts of the system
SysInfo	DataGroups	REPLICA – security groups authorizing access to shared data
SysInfo	GroupUsers	REPLICA – DataGroup user account membership
SysInfo	RoleMethods	REPLICA – the API methods assigned to each APIRole
SysInfo	RoleUsers	REPLICA – APIRole user account membership
SysMetrics	AutoWorkLog	STANDARD – log data written by AutoWork automated processes
SysMetrics	DGPErrors	STANDARD – exceptions, errors and info stored by a DGP system
SysMetrics	LatticeMetrics	STANDARD – production performance data stored by select users
SysMetrics	TestResults	STANDARD – test results saved by the API Tester test harness
SysWork	GeneralWork	STANDARD – count and duplicate checks for REPLICA tables
SysWork	ReplicaWork	STANDARD – replication and verification for REPLICA tables

Constant Field Values

The API field values are generally configuration values, types, states, and so on. The API messages are XML fragments and are not serialized objects or strongly typed. These constants help make it easier to call the API methods plus help standardize spelling and capitalization whenever they are used in the code.

```
public static class LocState
{
    public const string Online = "ONLINE";
    public const string Offline = "OFFLINE";
    public const string LocType = "LOCTYPE";
    public const string Source = "SOURCE";
    public const string Destination = "DESTINATION";
}

public static class Network
{
    public const string localhost = "LOCALHOST";
    public const string Internal = "INTERNAL";
    public const string DMZ = "DMZ";
    public const string External = "EXTERNAL";
}

public static class SchemaType
{
    public const string Replica = "REPLICA";
    public const string HybFile = "HYBFILE";
    public const string HybRec = "HYBREC";
}
```

```
public static class WorkType
{
    public const string Replicate = "REPLICATE";
    public const string Verify = "VERIFY";
    public const string DupeCheck = "DUPECHECK";
    public const string CountCheck = "COUNTCHECK";
}
```

```
public static class AcctType
{
    public const string User = "USER";
    public const string System = "SYSTEM";
}
```

```
public static class AcctState
{
    public const string Enabled = "ENABLED";
    public const string Disabled = "DISABLED";
}
```

```
public static class AccessLevel
{
    public const string ReadOnly = "READONLY";
    public const string ReadWrite = "READWRITE";
}
```

```
public static class APIData
{
    public const string Int = "INT";
    public const string Num = "NUM";
    public const string Text = "TEXT";
    public const string Date = "DATE";
    public const string XML = "XML";
    public const string JSON = "JSON";
    public const string DataTable = "DATATABLE";
}
```

```
public static class APIResult
{
    public const string OK = "OK";
    public const string Empty = "EMPTY";
    public const string Error = "ERROR";
    public const string Exception = "EXCEPTION";
    public const string Done = "DONE";
}
```

```
public static class APISetting
{
    public const string ON = "ON";
    public const string OFF = "OFF";
}
```

```
public static class BoolVal
{
    public const string TRUE = "TRUE";
    public const string FALSE = "FALSE";
}
```

```
public static class AuthState
{
    public const string OK = "OK";
    public const string Expired = "EXPIRED";
    public const string Error = "ERROR";
    public const string Nomatch = "NOMATCH";
    public const string Disabled = "DISABLED";
    public const string Duplicate = "DUPLICATE";
    public const string Exceeded = "EXCEEDED";
    public const string Content = "CONTENT";
    public const string Timeout = "TIMEOUT";
    public const string Missing = "MISSING";
    public const string Exception = "EXCEPTION";
}
```

```
public static class RunState
{
    public const string Ready = "READY";
    public const string Claimed = "CLAIMED";
    public const string Stopped = "STOPPED";
    public const string Disabled = "DISABLED";
}
```

```
public static class ProcState
{
    public const string Current = "CURRENT";
    public const string Working = "WORKING";
    public const string Complete = "COMPLETE";
    public const string Error = "ERROR";
}
```

```
public static class RecState
{
    public const string Active = "ACTIVE";
    public const string Edited = "EDITED";
    public const string Deleted = "DELETED";
    public const string Duplicate = "DUPLICATE";
}

public static class ReplicaAction
{
    public const string Insert = "INSERT";
    public const string Update = "UPDATE";
    public const string Delete = "DELETE";
    public const string Recover = "RECOVER";
}

public static class MethReturn
{
    public const string Default = "DEFAULT";
    public const string None = "NONE";
    public const string MethodError = "METHODERROR";
}

public static class SearchModeVal
{
    public const string ByFolder = "BYFOLDER";
    public const string ByMetadata = "BYMETADATA";
    public const string ByFavorite = "BYFAVORITE";
    public const string ByTags = "BYTAGS";
}
```


Constant Field Names

The constant field names are generally lists of the fields in the various mapper classes that accept API Request messages as inputs. Using these constants as if they were “string enums” makes it easier to call the API methods plus helps to standardize the spelling and capitalization of the various field names.

```
public static class CommonFields
{
    public const string row_id = "row_id";
    public const string src_id = "src_id";
    public const string row_ms = "row_ms";
    public const string src_ms = "src_ms";
    public const string rec_gid = "rec_gid";
    public const string rec_dbname = "rec_dbname";
    public const string rec_state = "rec_state";
    public const string rec_user = "rec_user";
    public const string start_rec_state = "start_rec_state";
    public const string ResplInfo = "ResplInfo";
    public const string SchemaFlag = "SchemaFlag";
    public const string PageNum = "PageNum";
    public const string PageSize = "PageSize";
    public const string SourceRecords = "SourceRecords";
    public const string Action = "Action";
}
```

```
public static class LogFields
{
    public const string CompName = "CompName";
    public const string AppName = "AppName";
    public const string FormName = "FormName";
    public const string ClassName = "ClassName";
    public const string MsgLoc = "MsgLoc";
    public const string ErrLevel = "ErrLevel";
    public const string ErrMessage = "ErrorMessage";
    public const string ErrData = "ErrData";
    public const string BatchCount = "BatchCount";
    public const string StartID = "StartID";
    public const string EndID = "EndID";
    public const string DurationMS = "DurationMS";
    public const string ProcState = "ProcState";
    public const string WebSvcName = "WebSvcName";
    public const string WebSvcVer = "WebSvcVer";
    public const string ClientTime = "ClientTime";
    public const string MethodCount = "MethodCount";
    public const string EndToEndMS = "EndToEndMS";
    public const string NetworkMS = "NetworkMS";
    public const string ServerMS = "ServerMS";
    public const string MsgName = "MsgName";
    public const string MsgData = "MsgData";
}
```

```
// *****  
// *****  
// DGPAdmin API  
// *****  
// *****
```

```
public static class APIMethodFields  
{  
    public const string MethodGID = "MethodGID";  
    public const string APIName = "APIName";  
    public const string MethodName = "MethodName";  
    public const string VersionName = "VersionName";  
    public const string MethodDescrip = "MethodDescrip";  
}
```

```
public static class APIRoleFields  
{  
    public const string RoleGID = "RoleGID";  
    public const string RoleName = "RoleName";  
    public const string RoleDescrip = "RoleDescrip";  
}
```

```
public static class DataGroupFields  
{  
    public const string GroupGID = "GroupGID";  
    public const string GroupName = "GroupName";  
    public const string GroupDescrip = "GroupDescrip";  
}
```

```
public static class GroupUserFields  
{  
    public const string GroupUserGID = "GroupUserGID";  
    public const string AccessLevel = "AccessLevel";  
}
```

```
public static class APIUserFields
{
    public const string UserID = "UserID";
    public const string Username = "Username";
    public const string Password = "Password";
    public const string FirstName = "FirstName";
    public const string MiddleName = "MiddleName";
    public const string LastName = "LastName";
    public const string Email = "Email";
    public const string AccountType = "AccountType";
    public const string AccountState = "AccountState";
    public const string ExpireDate = "ExpireDate";
    public const string MethodList = "MethodList";
    public const string ReadList = "ReadList";
    public const string WriteList = "WriteList";
    public const string MethodLimit = "MethodLimit";
    public const string SvcKeyVersion = "SvcKeyVersion";
}

public static class RoleMethodFields
{
    public const string RoleMethodGID = "RoleMethodGID";
}

public static class RoleUserFields
{
    public const string RoleUserID = "RoleUserID";
}
```

```
// *****  
// *****  
// DGPDrive API  
// *****  
// *****
```

```
public static class FolderFields
```

```
{  
    public const string FolderGID = "FolderGID";  
    public const string GroupGID = "GroupGID";  
    public const string ParentGID = "ParentGID";  
    public const string FolderName = "FolderName";  
    public const string DisplayOrder = "DisplayOrder";  
}
```

```
public static class FileFields
```

```
{  
    public const string FileGID = "FileGID";  
    public const string FileName = "FileName";  
    public const string FileDescrip = "FileDescrip";  
    public const string FileExt = "FileExt";  
    public const string FileSize = "FileSize";  
    public const string FileDate = "FileDate";  
    public const string FileHash = "FileHash";  
    public const string FileVersion = "FileVersion";  
    public const string FileServer = "FileServer";  
    public const string StoragePath = "StoragePath";  
    public const string SvcKeyVersion = "SvcKeyVersion";  
    public const string UploadFlag = "UploadFlag";  
    public const string SegSize = "SegSize";  
    public const string TotalSeg = "TotalSeg";  
    public const string CurrentSeg = "CurrentSeg";  
    public const string SegData = "SegData";  
    public const string SortBy = "SortBy";  
}
```

```
public const string SortOrder = "SortOrder";  
public const string Offset = "Offset";  
public const string ThreadID = "ThreadID";  
public const string EncKey = "EncKey";  
public const string EncFileSize = "EncFileSize";  
public const string FolderPath = "FolderPath";  
}
```

```
public static class TagFields  
{  
    public const string TagGID = "TagGID";  
    public const string TagName = "TagName";  
    public const string TagDescrip = "TagDescrip";  
    public const string FileTagGID = "FileTagGID";  
}
```

```
public static class FavoriteFields  
{  
    public const string FavoriteGID = "FavoriteGID";  
}
```

```
// *****  
// *****  
// TestReplica API  
// *****  
// *****  
  
public static class TestReplicaFields  
{  
    public const string DBName = "DBName";  
    public const string TestReplicaGID = "TestReplicaGID";  
    public const string Name = "Name";  
    public const string Value = "Value";  
}  
  
// *****  
// *****  
// DGPWork API  
// *****  
// *****  
  
public static class IntervalFields  
{  
    public const string MS = "MS";  
    public const string Sec = "SEC";  
    public const string MIN = "MIN";  
    public const string HOUR = "HOUR";  
    public const string TIMEOFDAY = "TIMEOFDAY";  
    public const string DAYOFWEEK = "DAYOFWEEK";  
    public const string DAYOFMONTH = "DAYOFMONTH";  
}
```

```
public static class WorkFields
{
    public const string WorkGID = "WorkGID";
    public const string WorkType = "WorkType";
    public const string SchemaType = "SchemaType";
    public const string SchemaTable = "SchemaTable";
    public const string SrcDBName = "SrcDBName";
    public const string NextRun = "NextRun";
    public const string ClaimedBy = "ClaimedBy";
    public const string ClaimTime = "ClaimTime";
    public const string Placeholder = "Placeholder";
    public const string StartID = "StartID";
    public const string EndID = "EndID";
    public const string FinalID = "FinalID";
    public const string FinalAction = "FinalAction";
    public const string BatchSize = "BatchSize";
    public const string IntervalMS = "IntervalMS";
    public const string RunState = "RunState";
    public const string StateMsg = "StateMsg";
    public const string SrcURL = "SrcURL";
    public const string DestURL = "DestURL";
    public const string SrcMethod = "SrcMethod";
    public const string DestMethod = "DestMethod";
    public const string Logging = "Logging";
    public const string ProcState = "ProcState";
    public const string InputParams = "InputParams";
    public const string IntervalType = "IntervalType";
    public const string IntervalVal = "IntervalVal";
    public const string MaxDurMS = "MaxDurMS";
    public const string ProcSteps = "ProcSteps";
    public const string Network = "Network";
}
```



```
public static class CountFields
{
    public const string SrcTblName = "SrcTblName";
    public const string MaxSrcID = "MaxSrcID";
    public const string MaxDestSrcID = "MaxDestSrcID";
    public const string SrcCount = "SrcCount";
    public const string SrcActiveCount = "SrcActiveCount";
    public const string DestSrcCount = "DestSrcCount";
    public const string DestSrcActiveCount = "DestSrcActiveCount";
}
```