

# Appendix A

---

## Diagnostics Log Files

### The PackageObj Log File

MTS Packages or COM+ Applications are a set of components that perform related functions. The MTS Server Packages or COM+ Applications are selected in the Packages and Detail node. This file contains information regarding each Package/Application activation process – each activation represents a single MTS Server Process or COM+ Application.

PackageObj		
Column	Datatype	Description
BeginDateTime	DATETIME	The date and time the server process started
BeginTime	FLOAT	System time (in milliseconds) when the server process started
ChkPtRecordID	CHAR(40)	Indicates this record is part of a ChkPt dump, and identifies the dump that it's associated with.
ComputerName	VARCHAR(225)	The name of the computer that this was created on
EndDateTime	DATETIME	The date and time the server process ended
EndTime	FLOAT	System time (in milliseconds) when the server process ended
GeneratedChkPtID	CHAR(40)	PackageActivations that crash generate a ChkPt record. This is the ID that is generated by this PackageActivation.
PackageGuid	CHAR(40)	The MTS package/COM+ application GUID
PackageName	VARCHAR(225)	The MTS package/COM+ application name
ProcessID	INT	The NT process ID
ShutdownReason	INT	0=Normal 1=Crash

**Table A-1 PackageObj Log File**

## The ActivityObj Log File

An Activity is a collection of MTS/COM+ objects that have a single logical thread of execution. This data is stored in the ActivityObj log file.

ActivityObj		
Column	Datatype	Description
Aborted	BOOLEAN	0=Normal 1=Aborted
ActivityGuid	CHAR(40)	MTS/COM+ activity GUID of the activity that started the transaction
BeginDateTime	DATETIME	The date and time when the activity started
BeginTime	FLOAT	System time (in milliseconds) when the activity started
ComputerName	VARCHAR(225)	The name of the computer that this activity was created on
EndDateTime	DATETIME	The date and time when the activity ended
EndTime	FLOAT	System time (in milliseconds) when the activity ended
FirstIID	CHAR(40)	Guid of first method to start in this activity
FirstMethod	INT	Interface ID of the first method to start in this activity
FirstObject	VARCHAR(255)	First object to start in this activity
IsDotNetObj	INT	Internal use only
NumActivities	INT	Internal use only
PackageGuid	CHAR(40)	The MTS package/COM+ application GUID
PackageName	VARCHAR(225)	The MTS package/COM+ application name
ThreadID	INT	The thread that started this activity
UserName	VARCHAR(225)	The name of the NT user associated with this activity

**Table A-2 ActivityObj Log File**

## The TxnObj Log File

Transactions represent the atomic units of work for an Activity. In terms of MTS and COM+, work starts and the Context SetCompletes or SetAborts. For transactional components, each Transaction results in a DTC transaction.

Transaction data is stored in the TxnObj log file. There is no primary key for this data.

TxnObj		
Column	Datatype	Description
ActivityGuid	CHAR(40)	MTS/COM+ activity GUID of the activity that started the transaction
BeginDateTime	DATETIME	The date and time the transaction started
BeginTime	FLOAT	System time (in milliseconds) when the transaction started
ChkPtRecordID	CHAR(40)	Indicates that this record is part of a ChkPt dump, as well as the dump that it's associated with
ComputerName	VARCHAR(225)	The name of the computer that this transaction was created on
EndDateTime	DATETIME	The date and time the transaction ended
EndTime	FLOAT	System time (in milliseconds) when the transaction ended
GeneratedChkPtID	CHAR(40)	ID of the ChkPt record that is generated by PackageActivations that crash or Transactions that abort
ObjectID	FLOAT	Instance ID of the component
PackageGuid	CHAR(40)	The MTS package/COM+ application GUID
PackageName	VARCHAR(225)	The MTS package/COM+ application name
TsdiProp	CHAR(40)	Internal use only
TxnGuidProp	CHAR(40)	The DTC Transaction Guid
TxnStateProp	INT	Outcome: 1 = Commit 2 = Abort 0 = Unknown
UniqueGuid	CHAR(40)	Unique identifier to link instances of activities, transactions, objects and methods

**Table A-3 TxnObj Log File**

## The ObjectObj Log File

An Object is a run time instance of a component. The Object data is recorded in the ObjectObj log file.

ObjectObj		
Column	Datatype	Description
Aborted	BOOLEAN	1=Aborted 0=Did not abort
ActivityGuid	CHAR(40)	MTS/COM+ activity GUID of the activity that started the transaction
BeginDateTime	DATETIME	The date and the object was created
BeginTime	FLOAT	System time (in milliseconds) when the object was created
ClientIP	CHAR(15)	W2K: Client IP address NT4: NULL
Clsid	VARCHAR(255)	Object name
Completed	BOOLEAN	1=Completed 0=Did not complete
ComputerName	VARCHAR(225)	The name of the computer that this object was created on
ContextID	FLOAT	W2K: ContextID of the component. NT4: NULL
CreatedTsidObj	INT	Internal use only
EndDateTime	DATETIME	The date and time the object completed
EndTime	FLOAT	System time (in milliseconds) when the object completed
ErrIID	CHAR(40)	Method GUID which caused exception
ErrMethod	INT	MethodID which caused exception
ErrStatus	INT	HRESULT of Methods in this object.
FirstIID	CHAR(40)	Guid of first method called
FirstMethod	INT	MethodID of first method called
IsFirstObject	INT	Internal use only
MethodExcept	INT	0 = Method Return 1 = Method Exception
NumResCreated	INT	Number of Resource Dispensers created
NumResUsed	INT	Number of Resource Dispensers used
ObjectID	FLOAT	Instance ID of the component

<b>ObjectObj</b>		
<b>Column</b>	<b>Datatype</b>	<b>Description</b>
PackageGuid	CHAR(40)	The MTS package/COM+ application GUID
PackageName	VARCHAR(225)	The MTS package/COM+ application name
ServerIP	CHAR(15)	W2K: Client IP address NT4: NULL
TotalResourceUsageTime	FLOAT	Internal use only
TsdiProp	CHAR(40)	Internal use only
TxnGuidProp	CHAR(40)	The DTC Transaction Guid
UniqueGuid	CHAR(40)	Unique identifier to link instances of activities, transactions, objects and methods
URL	VARCHAR (1024)	W2K: URL that created the object NT4: NULL

Table A-4 ObjectObj Log File

## The MethodObj Log File

Methods are procedures that act upon an object. The MethodObj log file contains every MethodCall on a component instance basis.

<b>MethodObj</b>		
<b>Columns</b>	<b>Datatype</b>	<b>Description</b>
ApplicationName	VARCHAR(255)	Name of the application
BeginDateTime	DATETIME	The date and time the method was called
BeginTime	FLOAT	System time (in milliseconds) when the method was called
ComputerName	VARCHAR(255)	The name of the computer where this method was called
EndDateTime	DATETIME	The date and time the method returned.
EndTime	FLOAT	System time (in milliseconds) when the method returned
ErrStatus	INT	HRESULT of the MethodCall
MethodID	INT	Interface ID for this method
MethodIID	CHAR(40)	GUID for this method
MethodName	VARCHAR(255)	Name of the method
ObjectID	FLOAT	Instance ID of the component

MethodObj		
Columns	Datatype	Description
UniqueGuid	CHAR(40)	Unique identifier to link instances of activities, transactions, objects and methods
Valid	INT	MTS misses events under heavy load so some data may be invalid. 0 = Some data may not be valid 1 = All data is valid

**Table A-5 MethodObj Log File**

## The ResourceObj Log File

ResourceDispenser is a service that provides the synchronization and management of non-durable resources within an object. The records are stored in the ResourceObj log file and they associate a ResourceDispenser with a particular instance.

ResourceObj		
Column	Datatype	Description
ActivityGuid	CHAR(40)	MTS/COM+ activity GUID of the activity that started the transaction
BeginDateTime	DATETIME	The date and time the resource dispenser started
BeginTime	FLOAT	System time (in milliseconds) when the resource dispenser started
ComputerName	VARCHAR(225)	The name of the computer that this resource dispenser was created on
EndDateTime	DATETIME	The date and time the resource dispenser ended
EndTime	FLOAT	System time (in milliseconds) when the resource dispenser ended
MaxAllocTime	FLOAT	Maximum time used
MinAllocTime	FLOAT	Minimum time used
NumAllocations	INT	Number of allocations
ObjectID	FLOAT	Instance ID of the component
PackageName	VARCHAR(225)	The MTS package/COM+ application name
ResourceID	INT	Reference to the component instance
ResType	VARCHAR(225)	Resource dispenser type
StartAllocTime	FLOAT	The time at which the resource dispenser became active
TotalAllocTime	FLOAT	The total time that the resource dispenser was active
UniqueGuid	CHAR(40)	Unique identifier to link instances of activities, transactions, objects and methods

**Table A-6 ResourceObj Log File**

## The ActtxnObj Log File

The ActtxnObj log file contains records that can be used to reconstruct the Activities and Transactions. Each record contains a UniqueGuid, ActivityGuid, and the begin and end times of the Activity. This UniqueGuid is also recorded in the TxnObj, ObjectObj, MethodObj, and ResourceObj log files.

ActtxnObj		
Column	Datatype	Description
ActivityGuid	CHAR(40)	MTS/COM+ activity GUID of the activity that started the transaction
BeginDateTime	DATETIME	The date and time the transaction started
BeginTime	FLOAT	System time (in milliseconds) when the transaction started
ChkPtRecordID	CHAR(40)	Indicates that this record is part of a ChkPt dump, as well as the dump that it is associated with
ComputerName	VARCHAR(225)	The name of the computer that this was created on
EndDateTime	DATETIME	The date and time the transaction ended
EndTime	FLOAT	System time (in milliseconds) when the transaction ended
FirstMethod	INT	Internal use only
FirstObject	VARCHAR(255)	Internal use only
ObjectID	FLOAT	Instance ID of the component
PackageGuid	CHAR(40)	The MTS package/COM+ application GUID
PackageName	VARCHAR(225)	The MTS package/COM+ application name
StartTxnPoint	INT	Internal use only
TsidProp	CHAR(40)	Internal use only
TxnGuidProp	CHAR(40)	The DTC Transaction Guid
UniqueGuid	CHAR(40)	Unique identifier to link instances of activities, transactions, objects and methods
UsesDTC	BOOLEAN	Indicates whether this transaction was part of a DTC transaction: 1=DTC transaction 0=Not DTC transaction

**Table A-7 ActtxnObj Log File**

## Checkpoint Records and Checkpoint Ids

During normal operation, many log records are generated. For TransactionAbort and PackageCrash error conditions, AppMetrics generates special records called Checkpoint records. These records are the same as other records, except that their GeneratedChkPtID value is non-NULL. The date and time, system time (in milliseconds), CrashType, and GeneratedChkPtID are also written to the ChkPt log file.

In addition to indicating which Transactions abort and which Packages crash, it also reports all the activities, transactions, components, methods, and resource dispensers that were active at the time of the error. The ChkPtRecordID in the ActtxnObj log file contains the same GeneratedChkPtID that was generated at the time of the error. The UniqueGuid in the ActtxnObj log file can then be used to find all entities that were active.

ChkPt		
Columns	Datatype	Description
BeginDateTime	DATETIME	The date and time when the error occurred
BeginTime	FLOAT	System time (in milliseconds) when the error occurred
ComputerName	VARCHAR(255)	The name of the computer where the error occurred
CrashType	INT	1=Transaction Abort 2=Package Crash
GeneratedChkPtID	CHAR(40)	ID of the ChkPt record that is generated for PackageActivations that crash or Transactions that abort

**Table A-8 Checkpoint Records**

## SQL Table Definitions for Diagnostics Template Log Files

The following table definitions show SQL syntax that is used to create a relational database with the data gathered via the monitors.

```
CREATE TABLE [ActivityObj] (  
  [Aborted] bit NULL,  
  [ActivityGuid] char (40) PRIMARY KEY,  
  [BeginDateTime] datetime NULL,  
  [BeginTime] float NULL,  
  [ComputerName] varchar (255) NULL,  
  [EndDateTime] datetime NULL,  
  [EndTime] float NULL,  
  [FirstIID] char (40) NULL,  
  [FirstMethod] int NULL,  
  [FirstObject] varchar (255) NULL,  
  [IsDotNetObj] bit NULL,  
  [NumActivities] int NULL,  
  [PackageGuid] char (40) NULL,  
  [PackageName] varchar (255) NULL,  
  [ThreadId] int NULL,  
  [UserName] varchar (255) NULL)
```

```
CREATE TABLE [ActtxnObj] (  
  [ActivityGuid] char (40),  
  [BeginDateTime] datetime NULL,  
  [BeginTime] float NULL,  
  [ChkPtRecordID] char(40) NULL,  
  [ComputerName] varchar (255) NULL,  
  [EndDateTime] datetime NULL,  
  [EndTime] float NULL,  
  [FirstMethod] int NULL,  
  [FirstObject] varchar (255) NULL,  
  [ObjectID] float NULL,  
  [PackageGuid] char (40) NULL,  
  [PackageName] varchar (255) NULL,  
  [StartTxnPoint] int NULL,  
  [TsidProp] char (40) NULL,  
  [TxnGuidProp] char (40) NULL,  
  [UniqueGuid] char (40),  
  [UsesDTC] bit NULL)
```

```
CREATE TABLE [ChkPt] (  
  [BeginDateTime] datetime NULL,  
  [BeginTime] float NULL,  
  [ComputerName] varchar (255) NULL,  
  [CrashType] int NULL,  
  [GeneratedChkPtID] char (40) PRIMARY KEY)
```

```
CREATE TABLE [MethodObj] (  
  [ApplicationName] char(255) NULL,  
  [BeginDateTime] datetime NULL,  
  [BeginTime] float NULL,
```

```
[ComputerName] varchar (255) NULL,  
[EndDateTime] datetime NULL,  
[EndTime] float NULL,  
[Errstatus] int NULL,  
[MethodID] int NULL,  
[MethodIID] char (40) NULL,  
[MethodName] varchar (255) NULL,  
[ObjectID] float NULL,  
[UniqueGuid] char (40),  
[Valid] int NULL)
```

```
CREATE TABLE [ObjectObj] (  
[Aborted] bit NULL,  
[ActivityGuid] char (40) NULL,  
[BeginDateTime] datetime NULL,  
[BeginTime] float NULL,  
[ClientIP] char(15), NULL,  
[clsid] varchar(255) NULL,  
[Completed] bit NULL,  
[ComputerName] varchar (255) NULL,  
[ContextID] float NULL,  
[CreatedTsidObj] int NULL,  
[EndDateTime] datetime NULL,  
[EndTime] float NULL,  
[ErrID] char (40) NULL,  
[ErrMethod] int NULL,  
[Errstatus] int NULL,  
[FirstIID] char (40) NULL,  
[FirstMethod] int NULL,  
[IsFirstObject] bit NULL,  
[MethodExcept] int NULL,  
[NumResCreated] int NULL,  
[NumResUsed] int NULL,  
[ObjectID] float,  
[PackageGuid] char (40) NULL,  
[PackageName] varchar (255) NULL,  
[ServerIP] char (15) NULL,  
[TotalResourceUsageTime] float NULL,  
[TsidProp] char (40) NULL,  
[TxnGuidProp] char (40) NULL,  
[UniqueGuid] char (40),  
[URL] [varchar] (1024) NULL)
```

```
CREATE TABLE [PackageObj] (  
[BeginDateTime] datetime NULL,  
[BeginTime] float NULL,  
[ChkPtRecordID] char (40) NULL,  
[ComputerName] varchar (255) NULL,  
[EndDateTime] datetime NULL,  
[EndTime] float NULL,  
[GeneratedChkPtID] char(40) NULL,  
[PackageGuid] char (40),  
[PackageName] varchar (255) NULL,  
[ProcessID] char (40) NULL,  
[ShutdownReason] int NULL)
```

```
CREATE TABLE [ResourceObj] (  
  [ActivityGuid] char (40) NULL,  
  [BeginDateTime] datetime NULL,  
  [BeginTime] float NULL,  
  [ComputerName] varchar (255) NULL,  
  [EndDateTime] datetime NULL,  
  [EndTime] float NULL,  
  [MaxAllocTime] float NULL,  
  [MinAllocTime] float NULL,  
  [NumAllocations] int NULL,  
  [ObjectID] float NULL,  
  [PackageName] varchar (255) NULL,  
  [ResourceID] int,  
  [ResType] varchar (255) NULL,  
  [StartAllocTime] float NULL,  
  [TotalAllocTime] float NULL,  
  [UniqueGuid] char (40))  
  
CREATE TABLE [TxnObj] (  
  [ActivityGuid] char (40) NULL,  
  [BeginDateTime] datetime NULL,  
  [BeginTime] float NULL,  
  [ChkPtRecordID] char(40) NULL,  
  [ComputerName] varchar (255) NULL,  
  [EndDateTime] datetime NULL,  
  [EndTime] float NULL,  
  [GeneratedChkPtID] char(40) NULL,  
  [ObjectID] float NULL,  
  [PackageGuid] char (40) NULL,  
  [PackageName] varchar (255) NULL,  
  [TsidProp] varchar (255) NULL,  
  [TxnGuidProp] char (40) PRIMARY KEY,  
  [TxnStateProp] int NULL,  
  [UniqueGuid] char (40))
```

**Figure A-1 SQL Table Definitions for Diagnostics**