Reports Monitor Reference

Monitor report execution and delivery.

Reports Monitor (Reference) provides real-time visibility into server-side report generation operations for ReportForms and WebData objects during runtime. The Reports Monitor displays:

- Report generation status
- File save operations
- Error tracking
- Execution results
- Server-side operations only

<ac:structured-macro ac:name="info"> ac:rich-text-body Only server-side operations are monitored. Client-side report methods are not included in the Designer monitoring tool. </ac:rich-text-body> </ac: structured-macro>

Access via **Reports** **Reports Monitor** when connected to runtime.

Monitor Table Properties
Runtime Property Access
Status Values
Monitoring Scope
Using Monitor Data
Best Practices Guidelines
Troubleshooting

Monitor Table Properties

Property	Description	Runtime Access
Name	Report identifier	`@Report.Form.<Name>`
Disable	Object disabled state	`@Report.Form.<Name>.Disable`
SavedFileName	Output file path	`@Report.Form.<Name>.SavedFileName`
LastStatus	Operation result (Success/Running/Error)	`@Report.Form.<Name>.LastStatus`
LastStatusMessage	Detailed error description	`@Report.Form.<Name>.LastStatusMessage`

Runtime Property Access

Access monitor properties in displays or scripts:

ReportForm Properties

csharp

```csharp
// Get report status
string status = @Report.Form.ProductionReport.LastStatus;
string fileName = @Report.Form.ProductionReport.SavedFileName;
string error = @Report.Form.ProductionReport.LastStatusMessage;

// Check for errors
if (@Report.Form.ProductionReport.LastStatus == "Error")
{
    LogError(@Report.Form.ProductionReport.LastStatusMessage);
}
```

WebData Properties

csharp

```csharp
// Monitor WebData operations
string apiStatus = @Report.WebData.APIReport.LastStatus;
bool isDisabled = @Report.WebData.APIReport.Disable;

// Validate execution
if (@Report.WebData.APIReport.LastStatus == "Success")
{
    ProcessReport();
}
```

Status Values

Status	Description	Action
Success	Report generated successfully	Process/distribute report
Running	Generation in progress	Wait for completion
Error	Generation failed	Check LastStatusMessage
Disabled	Object disabled	Enable or skip

Monitoring Scope

Server Operations (Monitored)

- Report generation triggered by server
- Scheduled report creation
- Server-side save operations
- Batch report processing

Client Operations (Not Monitored)

- Local report generation
- Client-side PDF creation
- Display-triggered reports
- User-initiated saves

Using Monitor Data

Health Monitoring

csharp

```csharp
// Check all reports health
public void CheckReportHealth()
{
    if (@Report.Form.DailyReport.LastStatus == "Error")
    {
        SendAlert("Daily Report Failed: " +
                  @Report.Form.DailyReport.LastStatusMessage);
    }
}
```

Batch Status Tracking

csharp

```csharp
// Monitor multiple reports
string[] reports = {"Daily", "Weekly", "Monthly"};
foreach(string report in reports)
{
    string status = @Report.Form[report].LastStatus;
    if (status != "Success")
    {
        // Handle failure
    }
}
```

Error Logging

csharp

```
// Log all report errors
public void LogReportErrors()
{
    if (@Report.Form.MyReport.LastStatus == "Error")
    {
        string error = string.Format(
            "{0}: Report {1} Error: {2}",
            DateTime.Now,
            "MyReport",
            @Report.Form.MyReport.LastStatusMessage
        );
        WriteToLog(error);
    }
}
```

Best Practices Guidelines

- [ ] **Monitor critical reports** - Check status regularly
- [ ] **Log all errors** - Track failure patterns
- [ ] **Set up alerts** - Notify on failures
- [ ] **Verify file creation** - Check SavedFileName
- [ ] **Handle running state** - Implement timeouts
- [ ] **Document issues** - Keep error history
- [ ] **Create dashboards** - Display report status

Troubleshooting

Report shows "Error":

- Check LastStatusMessage for details
- Verify file path permissions
- Review report configuration
- Check data availability

Status stuck on "Running":

- Check server resources
- Review report complexity
- Monitor database queries
- Implement timeout logic

No monitor data:

- Verify server-side execution
- Check Designer connection
- Confirm runtime active
- Review object configuration

File not created:

- Check SaveFileName path
- Verify write permissions
- Review disk space
- Check antivirus settings

In this section...
